

Veritas Access Administrator's Guide

Linux 7.3.2

(Use with the Veritas Access 3340
Appliance documentation)

Veritas Access Administrator's Guide

Last updated: 2018-08-10

Document version: 7.3.2 Rev 1

Legal Notice

Copyright © 2018 Veritas Technologies LLC. All rights reserved.

Veritas, the Veritas Logo, Veritas InfoScale, and NetBackup are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third-party software for which Veritas is required to provide attribution to the third party ("Third-Party Programs"). Some of the Third-Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the third-party legal notices document accompanying this Veritas product or available at:

<https://www.veritas.com/licensing/process>

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Veritas Technologies LLC and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. VERITAS TECHNOLOGIES LLC SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

<http://www.veritas.com>

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

<https://www.veritas.com/support>

You can manage your Veritas account information at the following URL:

<https://my.veritas.com>

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)

CustomerCare@veritas.com

Japan

CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The document version appears on page 2 of each guide. The latest documentation is available on the Veritas website:

<https://sort.veritas.com/documents>

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

doc.feedback@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

<http://www.veritas.com/community/>

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Section 1	Introducing Veritas Access	17
Chapter 1	Introducing Veritas Access	18
	About Veritas Access	18
	Accessing the Veritas Access CLI	24
	Navigating the Veritas Access CLI	24
	Getting help using the Veritas Access command-line interface	27
	Displaying the command history	29
	Using the more command	29
Section 2	Configuring Veritas Access	31
Chapter 2	Adding users or roles	32
	Adding Master, System Administrator, and Storage Administrator users	32
	About user roles and privileges	34
	About the naming requirements for adding new users	34
Chapter 3	Configuring the network	36
	About configuring the Veritas Access network	37
	About bonding Ethernet interfaces	37
	Bonding Ethernet interfaces	38
	Configuring DNS settings	40
	About the IP addresses for the Ethernet interfaces	42
	About Ethernet interfaces	42
	Displaying current Ethernet interfaces and states	43
	Configuring IP addresses	43
	Configuring Veritas Access to use jumbo frames	46
	Configuring VLAN interfaces	46
	Configuring NIC devices	47
	Swapping network interfaces	49
	Excluding PCI IDs from the cluster	50
	About configuring routing tables	51

	Configuring routing tables	52
	Changing the firewall settings	54
	IP load balancing	55
Chapter 4	Configuring authentication services	57
	About configuring LDAP settings	57
	Configuring LDAP server settings	58
	Administering the Veritas Access cluster's LDAP client	61
	Configuring the NIS-related settings	62
	Configuring NSS lookup order	63
Section 3	Managing Veritas Access storage	65
Chapter 5	Configuring storage	66
	About storage provisioning and management	67
	About configuring disks	67
	About configuring storage pools	68
	Configuring storage pools	68
	About quotas for usage	71
	Enabling, disabling, and displaying the status of file system quotas	72
	Setting and displaying file system quotas	73
	Setting user quotas for users of specified groups	75
	About quotas for CIFS home directories	75
	Workflow for configuring and managing storage using the Veritas Access CLI	77
	Displaying information for all disk devices associated with the nodes in a cluster	77
	Displaying WWN information	79
	Importing new LUNs forcefully for new or existing pools	80
	Initiating host discovery of LUNs	80
	Increasing the storage capacity of a LUN	81
	Formatting or reinitializing a disk	81
	Removing a disk	82
Chapter 6	Configuring data integrity with I/O fencing	83
	About I/O fencing	83
	Configuring disk-based I/O fencing	84
	Replacing an existing coordinator disk	85
	Disabling I/O fencing	85
	Destroying the coordinator pool	86

Chapter 7	Configuring iSCSI	87
	About iSCSI	87
	Configuring the iSCSI initiator	87
	Configuring the iSCSI initiator name	88
	Configuring the iSCSI devices	88
	Configuring discovery on iSCSI	89
	Configuring the iSCSI targets	92
	Modifying tunables for iSCSI	95
Chapter 8	Veritas Access as an iSCSI target	98
	About Veritas Access as an iSCSI target	98
	Managing the iSCSI targets	99
	Managing the iSCSI target service	101
	Managing the LUNs	102
	Managing the mappings with iSCSI initiators	105
	Managing the users	106
Section 4	Managing Veritas Access file access services	107
Chapter 9	Configuring your NFS server	108
	About using NFS server with Veritas Access	108
	Using the kernel-based NFS server	109
	Using the NFS-Ganesha server	109
	Switching between NFS servers	109
	Recommended tuning for NFS-Ganesha version 3 and version 4	110
	Accessing the NFS server	112
	Displaying and resetting NFS statistics	113
	Configuring Veritas Access for ID mapping for NFS version 4	114
	Configuring the NFS client for ID mapping for NFS version 4	115
	About authenticating NFS clients	115
	Setting up Kerberos authentication for NFS clients	115
	Adding and configuring Veritas Access to the Kerberos realm	116
Chapter 10	Using Veritas Access as a CIFS server	118
	About configuring Veritas Access for CIFS	119
	About configuring CIFS for standalone mode	120
	Configuring CIFS server status for standalone mode	121

Changing security settings	122
Changing security settings after the CIFS server is stopped	122
About Active Directory (AD)	122
Configuring entries for Veritas Access DNS for authenticating to Active Directory (AD)	122
Joining Veritas Access to Active Directory (AD)	124
Verifying that Veritas Access has joined Active Directory (AD) successfully	125
About configuring CIFS for Active Directory (AD) domain mode	125
Configuring CIFS for the AD domain mode	126
Using multi-domain controller support in CIFS	128
About leaving an AD domain	128
Changing domain settings for AD domain mode	129
Removing the AD interface	130
Setting NTLM	131
About setting trusted domains	132
Specifying trusted domains that are allowed access to the CIFS server	132
Allowing trusted domains access to CIFS when setting an IDMAP backend to rid	133
Allowing trusted domains access to CIFS when setting an IDMAP backend to ldap	135
Allowing trusted domains access to CIFS when setting an IDMAP backend to hash	136
Allowing trusted domains access to CIFS when setting an IDMAP backend to ad	137
About configuring Windows Active Directory as an IDMAP backend for CIFS	138
Configuring the Active Directory schema with CIFS-schema extensions	139
Configuring the LDAP client for authentication using the CLI	143
Configuring the CIFS server with the LDAP backend	143
Setting Active Directory trusted domains	144
About storing account information	145
Storing user and group accounts	147
Reconfiguring the CIFS service	149
About mapping user names for CIFS/NFS sharing	151
About the mapuser commands	152
Adding, removing, or displaying the mapping between CIFS and NFS users	153
Automatically mapping of UNIX users from LDAP to Windows users	154
About managing home directories	154

Setting the home directory file systems	155
Setting up home directories	156
Displaying home directory usage information	158
Deleting home directories and disabling creation of home directories	158
About CIFS clustering modes	159
About switching the clustering mode	160
About migrating CIFS shares and home directories	160
Migrating CIFS shares and home directories from normal to ctdb clustering mode	161
Migrating CIFS shares and home directories from ctdb to normal clustering mode	161
Setting the CIFS aio_fork option	162
About managing local users and groups	163
Creating a local CIFS user	163
Configuring a local group	164
Enabling CIFS data migration	165

Chapter 11	Configuring Veritas Access to work with Oracle Direct NFS	166
	About using Veritas Access with Oracle Direct NFS	166
	Best practices for improving Oracle database performance	167
	About the Oracle Direct NFS architecture	169
	About Oracle Direct NFS node or storage connection failures	170
	Configuring an Oracle Direct NFS storage pool	170
	Configuring an Oracle Direct NFS file system	171
	Configuring an Oracle Direct NFS share	173

Chapter 12	Configuring an FTP server	174
	About FTP	174
	Creating the FTP home directory	175
	Using the FTP server commands	175
	About FTP server options	176
	Customizing the FTP server options	179
	Administering the FTP sessions	180
	Uploading the FTP logs	180
	Administering the FTP local user accounts	181
	About the settings for the FTP local user accounts	182
	Configuring settings for the FTP local user accounts	183
	File sharing for a scale-out file system using FTP	185

Section 5	Managing the Veritas Access Object Store server	186
Chapter 13	Using Veritas Access as an Object Store server	187
	About the Object Store server	187
	Use cases for configuring the Object Store server	188
	Configuring the Object Store server	188
	About buckets and objects	192
	File systems used for objectstore buckets	194
	Multi-protocol support for NFS with S3	194
Section 6	Monitoring and troubleshooting	197
Chapter 14	Configuring event notifications and audit logs topics	198
	About event notifications	198
	About severity levels and filters	199
	About SNMP notifications	200
	Configuring an email group	200
	Configuring a syslog server	203
	Exporting events in syslog format to a given URL	204
	Displaying events on the console	205
	Configuring events for event reporting	205
	Configuring an SNMP management server	207
Section 7	Provisioning and managing Veritas Access file systems	209
Chapter 15	Creating and maintaining file systems	210
	About creating and maintaining file systems	210
	About scale-out file systems	211
	Erasure coding in a cluster file system (CFS) for NFS use case	214
	About encryption at rest	215
	Considerations for creating a file system	217
	Best practices for creating file systems	217
	Choosing a file system layout type	219
	Determining the initial extent size for a file system	220

About striping file systems	221
About creating a tuned file system for a specific workload	224
About FastResync	226
About scale-out fsck	226
Setting retention in files	227
Setting WORM over NFS	228
About managing application I/O workloads using maximum IOPS settings	228
Creating a file system	229
Bringing the file system online or offline	233
Listing all file systems and associated information	234
Modifying a file system	234
Adding or removing a mirror from a file system	234
Adding or removing a column from a file system	236
Increasing the size of a file system	237
Decreasing the size of a file system	239
Managing a file system	240
Defragmenting a file system	240
Checking and repairing a file system	243
Configuring FastResync for a file system	243
Disabling the FastResync option for a file system	244
Checking and resynchronizing stale mirrors	245
Setting file system alerts	246
Displaying file system alert values	247
Removing file system alerts	247
Destroying a file system	248
Upgrading disk layout versions	248

Section 8 Configuring cloud storage 250

Chapter 16 Configuring the cloud gateway 251

About the cloud gateway	251
Configuring the cloud gateway	252

Chapter 17 Configuring cloud as a tier 253

Configuring the cloud as a tier for scale-out file systems	253
Moving files between tiers in a scale-out file system	255
About policies for scale-out file systems	258
About pattern matching for data movement policies	259
About schedules for running policies	260
Creating and scheduling a policy for a scale-out file system	261

	Obtaining statistics on data usage in the cloud tier in scale-out file systems	265
Section 9	Provisioning and managing Veritas Access shares	268
Chapter 18	Creating shares for applications	269
	About file sharing protocols	269
	About concurrent access	270
	Sharing directories using CIFS and NFS protocols	271
	Sharing a file system as a CIFS home directory	274
	About concurrent access with NFS and S3	274
Chapter 19	Creating and maintaining NFS shares	275
	About NFS file sharing	275
	Displaying file systems and snapshots that can be exported	276
	Exporting an NFS share	276
	Displaying exported directories	280
	About managing NFS shares using netgroups	281
	Unexporting a directory or deleting NFS options	281
	Exporting an NFS share for Kerberos authentication	283
	Mounting an NFS share with Kerberos security from the NFS client	284
	Exporting an NFS snapshot	286
Chapter 20	Creating and maintaining CIFS shares	287
	About managing CIFS shares	288
	Exporting a directory as a CIFS share	288
	Configuring a CIFS share as secondary storage for an Enterprise Vault store	289
	Exporting the same file system/directory as a different CIFS share	289
	About the CIFS export options	290
	Setting share properties	294
	Hiding system files when adding a CIFS normal share	295
	Displaying CIFS share properties	295
	Allowing specified users and groups access to the CIFS share	296
	Denying specified users and groups access to the CIFS share	297
	Exporting a CIFS snapshot	298
	Deleting a CIFS share	298

	Modifying a CIFS share	299
	Making a CIFS share shadow copy aware	300
	Creating CIFS shares for a scale-out file system	300
Chapter 21	Using Veritas Access with OpenStack	302
	About the Veritas Access integration with OpenStack	302
	About the Veritas Access integration with OpenStack Cinder	303
	About the Veritas Access integration with OpenStack Cinder architecture	304
	Configuring Veritas Access with OpenStack Cinder	305
	About the Veritas Access integration with OpenStack Manila	312
	OpenStack Manila use cases	313
	Configuring Veritas Access with OpenStack Manila	314
	Creating a new share backend on the OpenStack controller node	315
	Creating an OpenStack Manila share type	316
	Creating an OpenStack Manila file share	317
	Creating an OpenStack Manila share snapshot	320
Section 10	Managing Veritas Access storage services	321
Chapter 22	Deduplicating data	322
	About data deduplication	322
	Best practices for using the Veritas Access deduplication feature	325
	Setting up deduplication	326
	Configuring deduplication	327
	Manually running deduplication	331
	Scheduling deduplication	333
	Setting deduplication parameters	336
	Removing deduplication	338
	Verifying deduplication	339
Chapter 23	Compressing files	341
	About compressing files	341
	About the compressed file format	342
	About the file compression attributes	342
	About the file compression block size	343
	Use cases for compressing files	343
	Best practices for using compression	343

Compression tasks	343
Compressing files	344
Showing the scheduled compression job	345
Scheduling compression jobs	345
Listing compressed files	346
Uncompressing files	346
Modifying the scheduled compression	347
Removing the specified schedule	348
Stopping the schedule for a file system	349
Removing the pattern-related rule for a file system	349
Removing the modified age related rule for a file system	349

Chapter 24 **Configuring SmartTier** 350

About Veritas Access SmartTier	351
How Veritas Access uses SmartTier	352
Configuring the policy of each tiered file system	354
Adding tiers to a file system	356
Adding or removing a column from a secondary tier of a file system	358
Configuring a mirror to a tier of a file system	359
Listing all of the files on the specified tier	361
Displaying a list of SmartTier file systems	361
About tiering policies	362
About configuring the policy of each tiered file system	362
Best practices for setting relocation policies	363
Relocating a file or directory of a tiered file system	364
Displaying the tier location of a specified file	365
About configuring schedules for all tiered file systems	365
Configuring schedules for tiered file systems	366
Displaying the files that may be moved or pruned by running a policy	367
Allowing metadata information on the file system to be written on the secondary tier	368
Restricting metadata information to the primary tier only	368
Removing a tier from a file system	368

Chapter 25 **Configuring SmartIO** 370

About SmartIO for solid-state drives	370
About configuring SmartIO	371
About SmartIO read caching for applications running on Veritas Access file systems	371
Setting up SmartIO read caching for Veritas Access	372

About SmartIO writeback caching for applications running on Veritas	
Access file systems	373
Tuning the writeback caching	374
Flushing dirty data from a writeback cache area	375
Setting up SmartIO writeback caching for VxFS file systems	376
Verifying the VxFS cache area and monitoring the caching	377
Setting the caching mode	377
Customizing the caching behavior	378
Viewing the caching statistics for a cache area	379

Chapter 26 Configuring episodic replication 380

About Veritas Access episodic replication	381
How Veritas Access episodic replication works	382
Starting Veritas Access episodic replication	383
Setting up communication between the source and the destination	
clusters	385
Setting up the file systems to replicate	389
Setting up files to exclude from an episodic replication unit	391
Scheduling the episodic replication	393
Defining what to replicate	395
About the maximum number of parallel episodic replication jobs	397
Managing an episodic replication job	397
Replicating compressed data	401
Displaying episodic replication job information and status	402
Synchronizing an episodic replication job	403
Behavior of the file systems on the episodic replication destination	
target	403
Accessing file systems configured as episodic replication destinations	
.....	404
Creating a recovery point objective (RPO) report	404
Episodic replication job failover and fallback	404
Process summary	406
Overview of the planned failover process	406
Overview of the planned fallback process	407
Overview of the unplanned failover process	407
Overview of the unplanned fallback process	408

Chapter 27 Configuring continuous replication 409

About Veritas Access continuous replication	409
How Veritas Access continuous replication works	410
How data flows in continuous replication synchronous mode	411

	How data flows in continuous replication asynchronous mode	413
	Starting Veritas Access continuous replication	415
	Setting up communication between the source and the destination clusters	417
	Setting up the file system to replicate	421
	Managing continuous replication	422
	Displaying continuous replication information and status	424
	Unconfiguring continuous replication	430
	Continuous replication failover and failback	431
	Process summary	432
	Overview of the planned failover process	432
	Overview of the planned failback process	433
	Overview of the unplanned failover process	433
	Overview of the unplanned failback process	433
Chapter 28	Using snapshots	435
	About snapshots	435
	Creating snapshots	436
	Displaying snapshots	437
	Managing disk space used by snapshots	438
	Bringing snapshots online or taking snapshots offline	440
	Restoring a snapshot	441
	About snapshot schedules	441
	Configuring snapshot schedules	442
	Managing automated snapshots	445
Chapter 29	Using instant rollbacks	448
	About instant rollbacks	448
	Creating a space-optimized rollback	450
	Creating a full-sized rollback	450
	Listing Veritas Access instant rollbacks	451
	Restoring a file system from an instant rollback	451
	Refreshing an instant rollback from a file system	452
	Bringing an instant rollback online	452
	Taking an instant rollback offline	452
	Destroying an instant rollback	453
	Creating a shared cache object for Veritas Access instant rollbacks	453
	Listing cache objects	455
	Destroying a cache object of a Veritas Access instant rollback	457

Chapter 30	Configuring Veritas Access with the NetBackup client	458
	About Veritas Access as a NetBackup client	459
	Prerequisites for configuring the NetBackup client	460
	About the NetBackup Snapshot Client	460
	About NetBackup snapshot methods	460
	Enabling or disabling the NetBackup SAN client	461
	Workflow for configuring Veritas Access for NetBackup	461
	Registering a NetBackup master server or adding an optional media server	463
	Displaying the excluded files from backup	464
	Displaying the included and excluded files for backups	464
	Adding or deleting patterns to the list of files in backups	465
	Configuring or resetting the virtual IP address used by NetBackup	466
	Configuring the virtual name of NetBackup	466
	Displaying the status of NetBackup services	467
	Configuring backup operations using NetBackup or other third-party backup applications	468
	Performing a backup or restore of a Veritas Access file system over a NetBackup SAN client	469
	Performing a backup or restore of a snapshot	470
	Installing or uninstalling the NetBackup client	470
	Configuring Veritas Access for NetBackup cloud storage	474
Section 11	Reference	477
Appendix A	Veritas Access documentation	478
	Using the Veritas Access product documentation	478
	About accessing the online man pages	479
Appendix B	Veritas Access tuning	481
	File system mount-time memory usage	481
Index		486

Introducing Veritas Access

- [Chapter 1. Introducing Veritas Access](#)

Introducing Veritas Access

This chapter includes the following topics:

- [About Veritas Access](#)
- [Accessing the Veritas Access CLI](#)
- [Navigating the Veritas Access CLI](#)
- [Getting help using the Veritas Access command-line interface](#)
- [Displaying the command history](#)
- [Using the more command](#)

About Veritas Access

Veritas Access is a software-defined scale-out network-attached storage (NAS) solution for unstructured data that works on commodity hardware. Veritas Access provides resiliency, multi-protocol access, and data movement to and from the public or private cloud based on policies.

You can use Veritas Access in any of the following ways.

Table 1-1 Interfaces for using Veritas Access

Interface	Description
GUI	Getting Started wizard with operations for managing the Veritas Access 3340 Appliance.
	Centralized dashboard with operations for managing your storage.
	See the GUI and the Online Help for more information.

Table 1-1 Interfaces for using Veritas Access (*continued*)

Interface	Description
RESTful APIs	Enables automation using scripts, which run storage administration commands against the Veritas Access cluster. See the <i>Veritas Access RESTful API Guide</i> for more information.
Command-line interface (CLI or CLISH)	Single point of administration for the entire cluster. See the manual pages for more information.

[Table 1-2](#) describes the features of Veritas Access.

Table 1-2 Veritas Access key features

Feature	Description
Multi-protocol access	Veritas Access includes support for the following protocols: <ul style="list-style-type: none">■ Amazon S3 See “About the Object Store server” on page 187.■ CIFS See “About configuring Veritas Access for CIFS” on page 119.■ FTP See “About FTP” on page 174.■ iSCSI target See “Configuring the iSCSI targets” on page 92.■ NFS See “About using NFS server with Veritas Access” on page 108.■ Oracle Direct NFS See “About using Veritas Access with Oracle Direct NFS” on page 166.■ SMB 3 See “About the CIFS export options” on page 290.■ NFS with S3 See “Multi-protocol support for NFS with S3” on page 194.
WORM storage for Enterprise Vault Archiving	Veritas Access can be configured as WORM primary storage for archival by Enterprise Vault. Veritas Access is certified as a CIFS primary WORM storage for Enterprise Vault 12.1. For more information, see the <i>Veritas Access Enterprise Vault Solutions Guide</i> .

Table 1-2 Veritas Access key features (*continued*)

Feature	Description
WORM support over NFS	Veritas Access supports WORM over NFS. See “Setting WORM over NFS” on page 228.
Creation of Partition Secure Notification (PSN) file for Enterprise Vault Archiving	A Partition Secure Notification (PSN) file is created at a source partition after the successful backup of the partition at the remote site. For more information, see the <i>Veritas Access Enterprise Vault Solutions Guide</i> .
Managing application I/O workloads using maximum IOPS settings	The MAXIOPS limit determines the maximum number of I/Os processed per second collectively by the storage underlying the file system. See “About managing application I/O workloads using maximum IOPS settings” on page 228.
Flexible Storage Sharing (FSS)	Enables cluster-wide network sharing of local storage.
Scale-out file system	The following functionality is provided for a scale-out file system: <ul style="list-style-type: none">■ File system that manages a single namespace spanning over both on-premises storage as well as cloud storage, which provides better fault tolerance for large data sets. See “About scale-out file systems” on page 211.■ Highly available NFS and S3 shares. You use scale-out file systems if you want to store a large capacity of data in a single namespace (3 PB is the maximum file system size). See “Using the NFS-Ganesha server” on page 109.■ Creation of CIFS shares. See “Creating CIFS shares for a scale-out file system” on page 300.■ File sharing for a scale-out file system using FTP. See “File sharing for a scale-out file system using FTP” on page 185.

Table 1-2 Veritas Access key features (*continued*)

Feature	Description
Cloud as a tier for a scale-out file system	<p>Veritas Access supports adding a cloud service as a storage tier for a scale-out file system. You can move data between the tiers based on file name patterns and when the files were last accessed or modified. Use scheduled policies to move data between the tiers on a regular basis.</p> <p>Veritas Access moves the data from the on-premises tier to Amazon S3, Amazon Glacier, Amazon Web Services (AWS), GovCloud (US), Azure, Google cloud, Alibaba, Veritas Access S3, IBM Cloud Object Storage, and any S3-compatible storage provider based on automated policies. You can also retrieve data archived in Amazon Glacier.</p> <p>See “Configuring the cloud as a tier for scale-out file systems” on page 253.</p>
SmartIO	<p>Veritas Access supports both read and writeback caching on solid state drives (SSDs) for applications running on Veritas Access file systems.</p> <p>See “About SmartIO for solid-state drives ” on page 370.</p>
SmartTier	<p>Veritas Access's built-in SmartTier feature can reduce the cost of storage by moving data to lower-cost storage. Veritas Access storage tiering also facilitates the moving of data between different drive architectures and on-premises.</p> <p>See “About Veritas Access SmartTier ” on page 351.</p>
Snapshot	<p>Veritas Access supports snapshots for recovering from data corruption. If files, or an entire file system, are deleted or become corrupted, you can replace them from the latest uncorrupted snapshot.</p> <p>See “About snapshots” on page 435.</p>
Deduplication	<p>You can run post-process periodic deduplication in a file system, which eliminates duplicate data without any continuous cost.</p> <p>See “About data deduplication” on page 322.</p>

Table 1-2 Veritas Access key features (*continued*)

Feature	Description
Compression	<p>You can compress files to reduce the space used, while retaining the accessibility of the files and having the compression be transparent to applications. Compressed files look and behave almost exactly like uncompressed files: the compressed files have the same name, and can be read and written as with uncompressed files.</p> <p>See “About compressing files” on page 341.</p>
Erasure-coding	<p>Erasure-coding is configured with the <i>EC log</i> option for NFS use case.</p> <p>See “Erasure coding in a cluster file system (CFS) for NFS use case” on page 214.</p>
IP load balancing	<p>With IP load balancing, a single virtual IP is used to act as a load balancer IP which distributes the incoming requests to the different nodes in the Veritas Access cluster for the services that are run on an active-active cluster.</p> <p>See “IP load balancing” on page 55.</p>
iSCSI target	<p>Veritas Access as an iSCSI target can be configured to serve block storage. iSCSI target as service is hosted in an active-active mode in the Veritas Access cluster.</p> <p>See “About Veritas Access as an iSCSI target” on page 98.</p>
NetBackup integration	<p>Built-in NetBackup client for backing up your file systems to a NetBackup master or media server. Once data is backed up, a storage administrator can delete unwanted data from Veritas Access to free up expensive primary storage for more data.</p> <p>See “About Veritas Access as a NetBackup client” on page 459.</p>
OpenStack plug-in	<p>Integration with OpenStack:</p> <ul style="list-style-type: none">■ OpenStack Cinder integration that allows OpenStack instances to use the storage hosted by Veritas Access. See “About the Veritas Access integration with OpenStack Cinder” on page 303.■ OpenStack Manila integration that lets you share Veritas Access file systems with virtual machines on OpenStack Manila. See “About the Veritas Access integration with OpenStack Manila” on page 312.

Table 1-2 Veritas Access key features (*continued*)

Feature	Description
Quotas	<p>Support for setting file system quotas, user quotas, and hard quotas.</p> <p>See “About quotas for usage” on page 71.</p>
Replication	<p>Periodic replication of data over IP networks.</p> <p>See “About Veritas Access episodic replication” on page 381.</p> <p>See the <code>episodic(1)</code> man page for more information.</p> <p>Synchronous replication of data over IP networks</p> <p>See “About Veritas Access continuous replication” on page 409.</p> <p>See the <code>continuous(1)</code> man page for more information.</p>
Support for LDAP, NIS, and AD	<p>Veritas Access uses the Lightweight Directory Access Protocol (LDAP) for user authentication.</p> <p>See “About configuring LDAP settings” on page 57.</p>
Partition Directory	<p>With support for partitioned directories, directory entries are redistributed into various hash directories. These hash directories are not visible in the name-space view of the user or operating system. For every new create, delete, or lookup, this feature performs a lookup for the respective hashed directory and performs the operation in that directory. This leaves the parent directory inode and its other hash directories unobstructed for access, which vastly improves file system performance.</p> <p>By default this feature is not enabled. See the <code>storage_fs(1)</code> manual page to enable this feature.</p>
Isolated storage pools	<p>Enables you to create an isolated storage pool with a self-contained configuration. An isolated storage pool protects the pool from losing the associated metadata even if all the configuration disks in the main storage pool fail.</p> <p>See “About configuring storage pools” on page 68.</p>

Table 1-2 Veritas Access key features (*continued*)

Feature	Description
Performance and tuning	<p>Workload-based tuning for the following workloads:</p> <ul style="list-style-type: none">■ Media server - Streaming media represents a new wave of rich Internet content. Recent advancements in video creation, compression, caching, streaming, and other content delivery technology have brought audio and video together to the Internet as rich media. You can use Veritas Access to store your rich media, videos, movies, audio, music, and photos.■ Virtual machine support See “About creating a tuned file system for a specific workload” on page 224.■ Other workloads

Accessing the Veritas Access CLI

To access the Veritas Access CLI

- 1 After installation, connect to the management console using the console IP address you assigned during the installation.
- 2 Log on to the management console by using the `admin` user.

User name: `admin`

Default password: `P@ssw0rd`

You are prompted to change your password after the initial logon.

Note: The End User License Agreement (EULA) is displayed the first time you log on to the Veritas Access CLI.

Navigating the Veritas Access CLI

All of the Veritas Access CLI commands are organized in different command modes depending on the operation you want to perform. You can get a list of the different command modes with descriptions of all the available modes by typing a question mark (?) at the CLI prompt.

To navigate the Veritas Access CLI

- 1 After logging on to the Veritas Access CLI, type a question mark (?) to see the available command modes.

You can see that you are in the `Storage>` mode because the cluster name now displays with the command mode.

`clustername.Storage`

```
-----  
|                               Veritas Access 7.3.2                               |  
|                                                                           |  
|                               Enterprise Edition                               |  
|                               Warning: Authorized Access Only                 |  
-----
```

```
Veritas Access 7.3.2 (Thu 23 Nov 11:09:05 PM PST 2017),  
Installed on Mon Nov 27 06:03:07 PST 2017
```

```
Welcome, master (Master). Today's date is Mon Dec 4 17:55:53 PST 2017
```

```
URL for accessing the GUI: https://10.182.11.171:14161/
```

```
smtf>
```

```
admin          -- Administrator user account  
backup         -- Backup configuration  
cifs           -- CIFS share commands  
cluster        -- Cluster configuration commands  
database       -- Database configuration  
exit           -- Return to the previous menus  
ftp            -- FTP configuration commands  
help           -- Display an overview of the CLI syntax  
history        -- Display command history  
logout         -- Logout of the current CLI session  
man            -- Display on-line reference manuals  
network        -- Network configuration commands  
nfs            -- NFS share commands  
objectaccess   -- Object Access commands  
openstack      -- Openstack configuration  
replication    -- Replication configuration  
report         -- Report utility commands  
SmartIO        -- SmartIO configuration  
storage        -- Storage provisioning commands
```

```
support          -- Support utility commands
system          -- System utility commands
upgrade         -- Software upgrade and version commands
```

- 2 Enter the `Storage>` mode by typing `storage` for example.

Getting help using the Veritas Access command-line interface

You can enter Veritas Access commands on the system console or from any host that can access Veritas Access through a session using Secure Shell (ssh).

Veritas Access provides the following features to help you when you enter commands on the command line:

- Auto-completion

The following keys both perform auto-completion for the current command line. If the command prefix is not unique, then the bell rings and a subsequent repeat of the key displays possible completions.

 - [enter] - Auto-completes, syntax-checks then executes a command. If there is a syntax error, then the offending part of the command line is highlighted and explained.
 - [space] - Auto-completes, or if the command is already resolved inserts a space.
- Command-line help

Type a question mark at the command line to display context-sensitive Help. This is either a list of possible command completions with summaries, or the full syntax of the current command. A subsequent repeat of this key, when a command has been resolved, displays a detailed reference.
- Keyboard shortcut keys

Move the cursor within the command line or delete text from the command line.
- Command-line manual pages

Type `man` and the name of the command.
- Error reporting

The ^ (caret) indicates a syntax error occurred in the preceding command statement. The location of a caret in the command statement indicates the location of the syntax error.
- Escape sequences

Substitute the command line for a previous entry.

Table 1-3 Conventions used in the Veritas Access online command-line man pages

Symbol	Description
(pipe)	Indicates you must choose one of elements on either side of the pipe.
[] (brackets)	Indicates that the element inside the brackets is optional.
{ } (braces)	Indicates that the element inside the braces is part of a group.
< >	Indicates a variable for which you need to supply a value.

Table 1-4 Veritas Access command-line keyboard shortcut keys for deletions

Shortcut key	Description
[CTRL-C]	Delete the whole line.
[CTRL-U]	Delete up to the start of the line from the current position.
[CTRL-W]	Delete one word to the left from the current position.
[ALT-D]	Delete one word to the right from the current position.
[CTRL-D]	Delete the character to the right on the insertion point.
[CTRL-K]	Delete all the characters to the right of the insertion point.
[CTRL-T]	Swap the last two characters.
[backspace]	Delete the character to the left of the insertion point.
[Del]	Delete one character from the current position.

Table 1-5 Escape sequences

Escape sequence	Description
!!	Substitute the last command line.
!N	Substitute the Nth command line (you can find the Nth command from using the <code>history</code> command).
!-N	Substitute the command line entered N lines before (the number is relative to the command you are entering).

Note: Most of the Veritas Access commands are executed asynchronously, so control may be returned to the command prompt before the operation is fully completed. For critical commands, you should verify the status of the command before proceeding. For example, after starting a CIFS resource, verify that the service is online.

Displaying the command history

The `history` command displays the commands that you have executed. You can also view commands executed by another user.

In addition to the commands that users execute with the CLISH, the `history` command displays internal commands that were executed by Veritas Access.

You must be logged in to the system to view the command history.

To display command history

- ◆ To display the command history, enter the following:

```
CLISH> history [username] [number_of_lines]
```

username Displays the command history for a particular user.

number_of_lines Displays the number of lines of history you want to view.

The information displayed from using the `history` command is:

Time	Displays the time stamp as MM-DD-YYYY HH:MM
Status	Displays the status of the command as Success, Error, or Warning.
Message	Displays the command description.
Command	Displays the actual commands that were executed by you or another user.

Using the more command

The `System> more` command enables, disables, or checks the status of the `more` filter. The default setting is enable, which lets you page through the text one screen at a time.

To modify and view the more filter setting

- ◆ To modify and view the `more` filter setting, enter the following:

```
System> more enable|disable|status
```

<code>enable</code>	Enables the more filter on all of the nodes in the cluster.
---------------------	-------------------------------------------------------------

<code>disable</code>	Disables the more filter on all of the nodes in the cluster.
----------------------	--------------------------------------------------------------

<code>status</code>	Displays the status of the <code>more</code> filter.
---------------------	------------------------------------------------------

Configuring Veritas Access

- [Chapter 2. Adding users or roles](#)
- [Chapter 3. Configuring the network](#)
- [Chapter 4. Configuring authentication services](#)

Adding users or roles

This chapter includes the following topics:

- [Adding Master, System Administrator, and Storage Administrator users](#)
- [About user roles and privileges](#)
- [About the naming requirements for adding new users](#)

Adding Master, System Administrator, and Storage Administrator users

The following administrator roles are included with Veritas Access:

- Master
- System Administrator
- Storage Administrator

You can add additional users with these roles. To add the different administrator roles, you must have `master` privilege.

Note: When adding a new user, you must assign a password.

To add a Master user

- ◆ Enter the following:

```
Admin> user add username master
```


To add a System Administrator user

- ◆ Enter the following:

```
Admin> user add username system-admin
```

To add a Storage Administrator user

- ◆ Enter the following:

```
Admin> user add username storage-admin
```

To change a user's password

- 1 Enter the following command to change the password for the current user:

```
Admin> passwd
```

You are prompted to enter your old password first. If the password matches, then you are prompted to enter the new password for the current user.

- 2 Enter the following command to change the password for a user other than the current user:

```
Admin> passwd [username]
```

You are prompted to enter your old password first. If the password matches, then you are prompted to enter the new password for the user.

To display a list of current users

- 1 Enter the following to display the current user:

```
Admin> show [username]
```

- 2 Enter the following to display a list of all the current users:

```
Admin> show
```

Enter the following to display the details of the administrator with the user name master:

```
Admin> show master>
```

To delete a user from Veritas Access

- 1 Enter the following if you want to display the list of all the current users before deleting a user:

```
Admin> show
```

- 2 Enter the following to delete a user from Veritas Access:

```
Admin> user delete username
```

Note: If the user does not get deleted normally, then you are prompted to delete the user forcefully.

About user roles and privileges

Your privileges within Veritas Access are based on what user role (Master, System Administrator, or Storage Administrator) you have been assigned.

The following table provides an overview of the user roles within Veritas Access.

Table 2-1 User roles within Veritas Access

User role	Description
Master	Masters are responsible for adding or deleting users, displaying users, and managing passwords. Only the Masters can add or delete other administrators.
System Administrator	System Administrators are responsible for configuring and maintaining the file system, NFS sharing, networking, clustering, setting the current date/time, and creating reports.
Storage Administrator	Storage Administrators are responsible for provisioning storage and exporting and reviewing reports.

About the naming requirements for adding new users

The following table provides the naming requirements for adding new Veritas Access users.

Table 2-2
Naming requirements for adding new users

Guideline	Description
Starts with	<p>Letter or an underscore (<code>_</code>)</p> <p>Must begin with an alphabetic character and the rest of the string should be from the following POSIX portable character set:</p> <p><code>([A-Za-z_][A-Za-z0-9_-]*[A-Za-z0-9_-.\$])</code>.</p>
Length	Can be up to 31 characters. If user names are greater than 31 characters, you will receive the error, "Invalid user name."
Case	Veritas Access CLI commands are case-insensitive (for example, the <code>user</code> command is the same as the <code>USER</code> command). However, user-provided variables are case-sensitive (for example, the username <code>Master1</code> is not the same as the username <code>MASTER1</code>).
Can contain	Hyphens (<code>-</code>) and underscores (<code>_</code>) are allowed.

Configuring the network

This chapter includes the following topics:

- [About configuring the Veritas Access network](#)
- [About bonding Ethernet interfaces](#)
- [Bonding Ethernet interfaces](#)
- [Configuring DNS settings](#)
- [About the IP addresses for the Ethernet interfaces](#)
- [About Ethernet interfaces](#)
- [Displaying current Ethernet interfaces and states](#)
- [Configuring IP addresses](#)
- [Configuring Veritas Access to use jumbo frames](#)
- [Configuring VLAN interfaces](#)
- [Configuring NIC devices](#)
- [Swapping network interfaces](#)
- [Excluding PCI IDs from the cluster](#)
- [About configuring routing tables](#)
- [Configuring routing tables](#)
- [Changing the firewall settings](#)
- [IP load balancing](#)

About configuring the Veritas Access network

Veritas Access has the following types of networks:

- Private network
The network between the nodes of the cluster itself. The private network is not accessible to Veritas Access client nodes.
- Public network
The public network is visible to all clients. Veritas Access uses static IP address for its public interface networking. Veritas Access does not support DHCP for public network configuration

Veritas Access supports the following operations to manage the networking settings:

- create bond
- vlan
- change IP addresses
- add or remove network interfaces
- swap or interchange existing network interfaces

About bonding Ethernet interfaces

Bonding associates a set of two or more Ethernet interfaces with one IP address. The association improves network performance on each Veritas Access cluster node by increasing the potential bandwidth available on an IP address beyond the limits of a single Ethernet interface. Bonding also provides redundancy for higher availability.

For example, you can bond two 1-gigabit Ethernet interfaces together to provide up to 2 gigabits per second of throughput to a single IP address. Moreover, if one of the interfaces fails, communication continues using the single Ethernet interface.

When you create a bond, you need to specify a bonding mode. In addition, for the following bonding modes: `802.3ad`, `balance-rr`, `balance-xor`, `broadcast`, `balance-tlb`, and `balance-alb`, make sure that the base network interface driver is configured correctly for the bond type. For type `802.3ad`, the switch must be configured for link aggregation.

Consult your vendor-specific documentation for port aggregation and switch set up. You can use the `-s` option in the Linux `ethtool` command to check if the base driver supports the link speed retrieval option. The `balance-alb` bond mode type works only if the underlying interface network driver enables you to set a link address.

Note: An added IPv6 address may go into a TENTATIVE state while bonding Ethernet interfaces with `balance-rr`, `balance-xor`, or `broadcast` bond modes. While bonding with those modes, Veritas Access requires the switch to balance incoming traffic across the ports, and not deliver looped back packets or duplicates. To work around this issue, enable EtherChannel on your switch, or avoid using these bond modes.

Table 3-1 Bonding mode

Index	Bonding mode	Fault tolerance	Load balancing	Switch setup	Ethtool/base driver support
0	balance-rr	yes	yes	yes	no
1	active-backup	yes	no	no	no
2	balance-xor	yes	yes	yes	no
3	broadcast	yes	no	yes	no
4	802.3ad	yes	yes	yes	yes (to retrieve speed)
5	balance-tlb	yes	yes	no	yes (to retrieve speed)
6	balance-alb	yes	yes	no	yes (to retrieve speed)

Note: When you create or remove a bond, SSH connections with Ethernet interfaces involved in that bond may be dropped. When the operation is complete, you must restore the SSH connections.

Bonding Ethernet interfaces

The `Network> bond create` and `Network> bond remove` operations involve bringing down the interface first and then bringing them back up. This may cause the SSH connections that are hosted over those interfaces to terminate. Use the physical console of the client rather than SSH when performing `Network> bond create` and `Network> bond remove` operations.

To display a bond

- ◆ To display a bond and the algorithm that is used to distribute traffic among the bonded interfaces, enter the following:

```
Network> bond show
```

To create a bond

- ◆ To create a bond between sets of two or more Ethernet interfaces on all Veritas Access cluster nodes, enter the following:

```
Network> bond create interfacelist mode option
```

interfacelist	Specifies a comma-separated list of public Ethernet interfaces to bond.
mode	Specifies how the bonded Ethernet interfaces divide the traffic.
option	Specifies a comma-separated option string. Available only when the bond mode is 2 (balance-xor) or 4 (802.3ad) xmit_hash_policy - specifies the transmit hash policy to use for slave selection in balance-xor and 802.3ad modes.

If the option is not specified correctly, you get an error.

You can specify a mode either as a number or a character string, as follows:

0	balance-rr	This mode provides fault tolerance and load balancing. It transmits packets in order from the first available slave through the last.
1	active-backup	Only one slave in the bond is active. If the active slave fails, a different slave becomes active. To avoid confusing the switch, the bond's MAC address is externally visible on only one port (network adapter).
2	balance-xor	Transmits based on the selected transmit hash policy. The default policy is a simple. This mode provides load balancing and fault tolerance.
3	broadcast	Transmits everything on all slave interfaces and provides fault tolerance.

4	802.3ad	Creates aggregation groups with the same speed and duplex settings. It uses all slaves in the active aggregator based on the 802.3ad specification.
5	balance-tlb	Provides channel bonding that does not require special switch support. The outgoing traffic is distributed according to the current load (computed relative to the speed) on each slave. The current slave receives incoming traffic. If the receiving slave fails, another slave takes over its MAC address.
6	balance-alb	Includes balance-tlb plus Receive Load Balancing (RLB) for IPV4 traffic. This mode does not require any special switch support. ARP negotiation load balances the receive.

To remove a bond

- ◆ To remove a bond from all of the nodes in a cluster, enter the following:

```
Network> bond remove bondname
```

where *bondname* is the name of the bond configuration.

Private bonding

- 1 The `Network> bond priv-create` command creates the bond for the private interfaces (priveth0 and priveth1) with mode 0 (*balance-rr*). Veritas Access supports only mode 0 for the private interfaces. To get the advantage of private network bonding, all private interfaces must be connected by a switch or a hub. The switch or hub ensures that if one of the NIC goes down, the communication continues with the other NIC. All services are brought offline before a bonded interface is created. This command has to be run using the server console.
- 2 The `Network> bond priv-remove` command removes the bonding of private interfaces (priveth0 and priveth1) for the cluster. All services are brought offline before a bonded interface is removed. This command has to be run using the server console.

Configuring DNS settings

The Domain Name System (DNS) service resolves names to IP addresses. You can configure Veritas Access to use DNS to look up domain names and IP addresses. You enable the DNS service for the cluster, then specify up to three DNS servers.

To display DNS settings

- ◆ To display DNS settings, enter the following:

```
Network> dns show
```

To enable DNS service

- ◆ To enable Veritas Access hosts to do DNS lookups, enter the following commands:

```
Network> dns enable
```

You can verify using the `dns show` command.

To disable DNS settings

- ◆ To disable DNS settings, enter the following:

```
Network> dns disable
```

You can verify using the `dns show` command.

To specify the IP addresses of the DNS name servers

- ◆ To specify the IP addresses of the DNS name servers used by the Veritas Access DNS service, enter the following commands:

```
Network> dns set nameservers nameserver1 [nameserver2] [nameserver3]
```

You can verify using the `dns show` command.

To remove the name servers list used by DNS

- ◆ To remove the name servers list used by DNS, enter the following commands:

```
Network> dns clear nameservers
```

You can verify using the `dns show` command.

To set the domain name for the DNS server

- ◆ To set the domain name for the DNS server, enter the following:

```
Network> dns set domainname domainname
```

where *domainname* is the domain name for the DNS server.

You can verify using the `dns show` command.

To allow multiple DNS search domains

- ◆ To allow multiple DNS search domains, enter the following:

```
Network> dns set searchdomains searchdomain1[,searchdomain2]  
[,searchdomain3]
```

where *searchdomain1* is the first DNS search domain to be searched. Specify the search domains in the order in which the search domains should be used.

To configure multiple DNS search domains that have already been entered

- ◆ To configure multiple DNS search domains that have already been entered, add the existing domain name with the new domain name as comma-separated entries.

```
Network> dns set searchdomains domain1.access.com,  
domain2.access.com.
```

You can verify using the `dns show` command.

To remove the domain name used by DNS

- ◆ To remove the domain name used by DNS, enter the following:

```
Network> dns clear domainname
```

You can verify using the `dns show` command.

About the IP addresses for the Ethernet interfaces

Internet Protocol (IP) commands configure your routing tables, Ethernet interfaces, and IP addresses, and display the settings.

To configure the Ethernet interfaces:

See [“About Ethernet interfaces”](#) on page 42.

About Ethernet interfaces

Each Ethernet interface must have a physical IP address associated with it. These are usually supplied when the Veritas Access software is installed.

Each Ethernet interface can be configured with a virtual IP address for clustering purposes in Veritas Access. This does not imply that each interface must have a virtual IP to communicate with the network.

The physical address must be present before adding a virtual address. To add an IPv6 address on an IPv4 cluster, you have to configure the IPv6 physical address and then add the virtual address for the given interface.

Displaying current Ethernet interfaces and states

To display current Ethernet interfaces and states

- ◆ To display current configurations, enter the following:

```
Network> ip link show [nodename] [device]
```

nodename Specifies which node of the cluster to display the attributes.

Enter `all` to display all the IP links.

device Specifies which Ethernet interface on the node to display the attributes.

To display all configurations, enter the following:

```
Network> ip link show
```

Configuring IP addresses

During installation, you specified a range of public IP addresses to be used for physical interfaces. You also specified a range for virtual interfaces. You can see which of these addresses are assigned to each node. You can use this procedure to verify the IP addresses in your configuration. You can add additional IP addresses if you want to add additional nodes and no other IP addresses are available.

To display all the IP addresses for the cluster

- ◆ To display all of a cluster's IP addresses, enter the following:

```
Network> ip addr show
```

The output headings are:

IP Displays the IP addresses for the cluster.

Netmask Displays the netmask for the IP address. Netmask is used for IPv4 addresses.

Specify an IPv4 address in the format AAA.BBB.CCC.DDD, where each number ranges from 0 to 255.

Prefix	Displays the prefix used for IPv6 addresses. The value is an integer in the range 0-128.
Device	Displays the name of the Ethernet interface for the IP address.
Node	Displays the node name associated with the interface.
Type	Displays the type of the IP address: physical or virtual.
Status	Displays the status of the IP addresses: <ul style="list-style-type: none"> ■ ONLINE ■ ONLINE (console IP) ■ OFFLINE ■ FAULTED <p>A virtual IP can be in the FAULTED state if it is already being used. It can also be in the FAULTED state if the corresponding device is not working on all nodes in the cluster (for example, a disconnected cable).</p>

To add an IP address to a cluster

- ◆ To add an IP address to a cluster, enter the following:

```
Network> ip addr add ipaddr netmask | prefix type
[device] [nodename]
```

ipaddr	Specifies the IP address to add to the cluster. <p>Do not use physical IP addresses to access the Veritas Access cluster. In case of failure, the IP addresses cannot move between nodes. A failure could be either a node failure, an Ethernet interface failure, or a storage failure.</p> <p>You can specify either an IPv4 address or an IPv6 address.</p>
netmask	Specifies the netmask for the IP address. Netmask is used for IPv4 addresses.
prefix	Specifies the prefix for the IPv6 address. The accepted range is 0-128 integers.
type	Specifies the IP address type, either virtual or physical. <p>If type is virtual, the device is used to add new IP address on that device.</p> <p>If type is physical, the IP address gets assigned to given node on given device. In this case, you have to specify the nodename.</p>

device	Only use this option if you entered <code>virtual</code> for the <i>type</i> .
nodename	Any node of the cluster

To change an IP address to online on a specified node

- ◆ To change an IP address to online on a specified node, enter the following:

```
Network> ip addr online ipaddr nodename
```

ipaddr	Specifies the IP address that needs to be brought online. You can specify either an IPv4 address or an IPv6 address.
nodename	Specifies the nodename on which the IP address needs to be brought online. If you do not want to enter a specific nodename, enter <i>any</i> with the IP address.

To modify an IP address

- ◆ To modify an IP address, enter the following:

```
Network> ip addr modify oldipaddr newipaddr netmask | prefix
```

oldipaddr	Specifies the old IP address to be modified, as either an IPv4 address or an IPv6 address. The specified <i>oldipaddr</i> must be assigned to the cluster.
newipaddr	Specifies the new IP address, as either an IPv4 address or an IPv6 address. The new IP address must be available.
netmask	Specifies the netmask for the new IP address. Netmask is used for IPv4 addresses.
prefix	Specifies the prefix for the IPv6 address. The value is an integer in the range 0-128.

To remove an IP address from the cluster

- ◆ To remove an IP address from the cluster, enter the following:

```
Network> ip addr del ipaddr
```

where *ipaddr* is either an IPv4 address or an IPv6 address.

Configuring Veritas Access to use jumbo frames

You can display and change the public Ethernet interfaces (for example, `pubeth0` and `pubeth1`) whether a link is up or down, and the Ethernet interface's Maximum Transmission Unit (MTU) value.

The MTU value controls the maximum transmission unit size for an Ethernet frame. The standard maximum transmission unit size for Ethernet is 1500 bytes (without headers). In supported environments, the MTU value can be set to larger values up to 9000 bytes. Setting a larger frame size on an interface is commonly referred to as using jumbo frames. Jumbo frames help reduce fragmentation as data is sent over the network and in some cases, can also provide better throughput and reduced CPU usage. To take advantage of jumbo frames, the Ethernet cards, drivers, and switching must all support jumbo frames.

Configuring VLAN interfaces

The virtual LAN (VLAN) feature lets you create VLAN interfaces on the Veritas Access nodes and administer them as any other VLAN interfaces. The VLAN interfaces are created using Linux support for VLAN interfaces.

Use the `Network> vlan` commands to view, add, or delete VLAN interfaces.

Note: To use VLAN, your network must have VLAN-supported switches.

To display the VLAN interfaces

- ◆ To display the VLAN interfaces, enter the following:

```
Network> vlan show
```

To add a VLAN interface

- ◆ To add a VLAN interface, enter the following:

```
Network> vlan add device vlan_id
```

<code>device</code>	Specifies the VLAN interface on which the VLAN interfaces will be added.
---------------------	--------------------------------------------------------------------------

<code>vlan_id</code>	Specifies the VLAN ID which the new VLAN interface uses. Valid values range from 1 to 4095.
----------------------	---------------------------------------------------------------------------------------------

To delete a VLAN interface

- ◆ To delete a VLAN interface, enter the following:

```
Network> vlan del vlan_device
```

where the *vlan_device* is the VLAN name from the `Network> vlan show` command.

Configuring NIC devices

To list NIC devices on a specified node

- ◆ To list NIC devices on a specified node, enter the following:

```
Network> device list nodename
```

where *nodename* is the specified node for which bus IDs and MAC addresses for all devices are listed.

To add a NIC device to a Veritas Access cluster

- ◆ To add a NIC device to a Veritas Access cluster, enter the following:

```
Network> device add devicename
```

where *devicename* is the name of the device that you want to add.

When any `eth#` device is added, the `eth#` device gets a new `pubeth#` in the Veritas Access cluster.

To remove a NIC device from a Veritas Access cluster

- ◆ To remove a NIC device from a Veritas Access cluster, enter the following:

```
Network> device remove devicename
```

where *devicename* is the name of the device you want to remove.

When a device is removed, all the physical IP addresses and virtual IP addresses that are associated with the device are deleted from the specified NIC device. All physical IP addresses are kept in a free list and will be available for reuse; virtual IP addresses are not available for reuse. You need to re-add the NIC device in cases of reuse.

You can use the `Network> ip addr show` command to display the list of IP addresses associated with the device. You can see an `UNUSED` status beside the IP addresses that are free (not used).

To rename a NIC device

- ◆ To rename a NIC device, enter the following:

```
Network> device rename old_name with new_name nodename
```

Only devices with the prefix `eth` can be renamed. NIC devices with new names should not be present on all nodes of the Veritas Access cluster. In cases of mismatches in names of newly-added NICs in the Veritas Access cluster, you can rename those devices, and then add the devices to the Veritas Access cluster.

To identify a NIC device

- ◆ To identify a NIC device, enter the following:

```
device identify devicename nodename [timeout]
```

`devicename` Specify the name of the device you want to identify.

`nodename` Specify the node on which the device is located.

`timeout` By default, the timeout value is 120 seconds.

To replace a NIC device from a Veritas Access cluster

- 1 Delete all the VIP addresses that are related to the NIC that you want to replace using the `ippr addr del` command. Enter the following:

```
Network> ip addr del <virtual IP>
```

- 2 Find out the name that is related to the NIC that is to be replaced by using the `device list` command.
- 3 Remove the device from the Veritas Access configuration using the `device remove` command
- 4 Shut down the target node and replace the target NIC hardware. Then restart the system.
- 5 If new NIC name is not the same as the original device name, rename the new device name to the original device name.

- 6 Add the new NIC device.

```
Network> device add <NIC device>
```

- 7 Add the VIP back to the device.

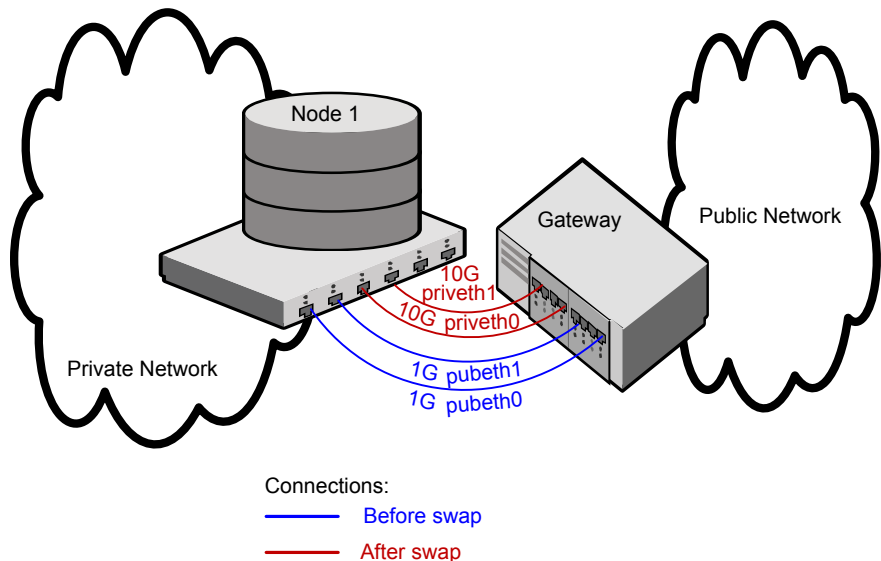
```
Network> ip addr add <virtual IP>
```

Swapping network interfaces

The `Network> swap` command can be used for swapping two network interfaces of a node in a cluster. This command helps set up the cluster properly in cases where the first node of a cluster cannot be pinged.

[Figure 3-1](#) describes a scenario whereby using the `Network> swap` command, you can use the more powerful 10G network interfaces to carry the public network load.

Figure 3-1 Scenario for using `Network> swap` for network interfaces



A System Administrator can use the `Network> swap` command in the following ways:

- Multi-node cluster: You can swap one public interface with another.
- Single-node cluster: You can swap a private interface with a public interface, swap two public interfaces, or swap two private interfaces.

If input to the `Network> swap` command contains one public and one private interface, and there are two separate switches for the private and the public network, then before you run the `Network> swap` command, the System Administrator has to exchange cable connections between these interfaces.

Running the `Network> swap` command requires stopping the given interfaces, which causes the following:

- After you run the `Network> swap` command, all SSH connection(s) hosted on the input interfaces terminate.
- If a public interface is involved when issuing the `Network> swap` command, all Virtual IP addresses (VIPs) hosted on that interface are brought down first, and are brought back up after `Network> swap` is complete.
- If the `Network> swap` command is run remotely, due to SSH connection termination, its end status may not be visible to the end user. You can check the status of the `Network> swap` command under `history`, by reconnecting to the cluster.

Note: Veritas Access recommends not to use the `Network> swap` command when active I/O load is present on the cluster.

To use the swap command

- ◆ To use the `Network> swap` command, enter the following:

```
Network> swap interface1 interface2 [nodename]
```

interface1 Indicates the name of the first network interface.

interface2 Indicates the name of the second network interface.

nodename Indicates the name of the node. If *nodename* is not provided, the `Network> swap` command is executed on the current node in the cluster.

Excluding PCI IDs from the cluster

During the initial Veritas Access software installation on the first node, you can exclude certain PCI IDs in your cluster to reserve them for future use. You may want to exclude additional PCD IDs when you add additional nodes to the cluster. You can add the PCI IDs to the exclusion list. The interface cards for which PCI ID's have been added in the PCI exclusion list are not used as private or public

interfaces for the subsequent cluster node install. During a new node install, the remaining PCI bus interfaces are searched and added as public or private interfaces.

The `Network> pciexclusion` command can be used with different options:

- The `Network> pciexclusion show` command displays the PCI IDs that have been selected for exclusion. It also provides information about whether it has been excluded or not by displaying y(yes) or n(no) symbols corresponding to the node name. If the node is in the INSTALLED state, it displays the UUID of the node.
- The `Network> pciexclusion add pcilist` command allows an administrator to add specific PCI ID(s) for exclusion. These values must be provided before the installation. The command excludes the PCI from the second node installation.
pcilist is a comma-separated list of PCI IDs.
- The `Network> pciexclusion delete pci` command allows an administrator to delete a given PCI ID from exclusion. This command must be used before the installation for it to take effect. The command is effective for the next node install
The *PCI* ID bits format is hexadecimal (XXXX:XX:XX.X).

About configuring routing tables

Sometimes a Veritas Access cluster must communicate with network services (for example, LDAP) using specific gateways in the public network. In these cases, you must define routing table entries.

These entries consist of the following:

- The target network node's IP address and accompanying netmask.
- Gateway's IP address.
- Optionally, a specific Ethernet interface via which to communicate with the target. This is useful, for example, if the demands of multiple remote clients are likely to exceed a single gateway's throughput capacity.

Configuring routing tables

To display the routing tables of the nodes in the cluster

- ◆ To display the routing tables of the nodes in the cluster, enter the following:

```
Network> ip route show [nodename]
```

where *nodename* is the node whose routing tables you want to display. To see the routing table for all of the nodes in the cluster, enter `all`.

For example:

```
Network> ip route show all
```

Destination	Displays the destination network or destination host for which the route is defined.
Gateway	Displays a network node equipped for interfacing with another network.
Genmask	Displays the netmask.
Flags	The flags are as follows: U - Route is up H - Target is a host G - Use gateway
MSS	Displays maximum segment size. The default is 0. You cannot modify this attribute.
Window	Displays the maximum amount of data the system accepts in a single burst from the remote host. The default is 0. You cannot modify this attribute.
irrt	Displays the initial round trip time with which TCP connections start. The default is 0. You cannot modify this attribute.
Iface	Displays the interface. On UNIX systems, the device name <code>lo</code> refers to the loopback interface.

To add to the route table

- ◆ To add a route entry to the routing table of nodes in the cluster, enter the following:

```
Network> ip route add nodename ipaddr netmask  
| prefix via gateway [dev device]
```

nodename	<p>Specifies the node to whose routing table the route is to be added.</p> <p>To add a route path to all the nodes, use <code>all</code> in the <i>nodename</i> field.</p> <p>If you enter a node that is not a part of the cluster, an error message is displayed.</p>
ipaddr	<p>Specifies the destination of the IP address.</p> <p>You can specify either an IPv4 address or an IPv6 address.</p> <p>If you enter an invalid IP address, then a message notifies you before you fill in other fields.</p>
netmask	<p>Specifies the netmask associated with the IP address that is entered for the <i>ipaddr</i> field.</p> <p>Use a netmask value of 255.255.255.255 for the netmask to add a host route to <i>ipaddr</i>.</p>
prefix	<p>Specifies the prefix for the IPv6 address. Accepted ranges are 0-128 integers.</p>
via	<p>This is a required field. You must type in the word.</p>
gateway	<p>Specifies the gateway IP address used for the route.</p> <p>If you enter an invalid gateway IP address, then an error message is displayed.</p> <p>To add a route that does not use a gateway, enter a value of 0.0.0.0.</p>
dev	<p>Specifies the route device option. You must type in the word.</p>
device	<p>Specifies which Ethernet interface on the node the route path is added to. This variable is optional.</p> <p>You can specify the following values:</p> <ul style="list-style-type: none"> ■ any - Default ■ pubeth0 - Public Ethernet interface ■ pubeth1 - Public Ethernet interface <p>The Ethernet interface field is required only when you specify <i>dev</i> in the <i>dev</i> field.</p> <p>If you omit the <i>dev</i> and <i>device</i> fields, Veritas Access uses a default Ethernet interface.</p>

To delete route entries from the routing tables of nodes in the cluster

- ◆ To delete route entries from the routing tables of nodes in the cluster, enter the following:

```
Network> ip route del nodename ipaddr  
          netmask | prefix
```

<i>nodename</i>	Specify the node from which the route is deleted. To delete the route entry from all nodes, use the <code>all</code> option in this field.
<i>ipaddr</i>	Specifies the destination IP address of the route entry to be deleted. You can specify either an IPv4 address or an IPv6 address. If you enter an invalid IP address, a message notifies you before you enter other fields.
<i>netmask</i>	Specifies the IP address to be used. Netmask is used for IPv4 addresses.
<i>prefix</i>	Specifies the prefix for the IPv6 address. Accepted ranges are 0-128 integers.

Changing the firewall settings

The `Network> firewall` command is used to view or change the firewall settings.

The firewall setting is enabled by default. If you want to allow connections on any port from any IP, you have to disable the firewall setting. Applied rules do not work when the firewall setting is disabled.

```
# firewall disable
```

The firewall setting can be enabled again to allow specific IPs to connect to the ports while blocking other connections.

```
# firewall enable
```

The current status of the firewall setting (whether it is enabled or disabled) can be displayed using the `firewall status` command.

```
# firewall status
```

See the `Network> firewall man` page for detailed examples.

IP load balancing

IP load balancing has been introduced in Veritas Access 7.3.1 release.

The purpose of this feature is to reduce the number of virtual IPs required for Veritas Access. With IP load balancing, a single virtual IP is used to act as a load balancer IP which distributes the incoming requests to the different nodes in the Veritas Access cluster for the services that are run on an active-active cluster.

Note: A request is a client session and not an individual I/O. Therefore, the IP load balancing feature balances the incoming client sessions and distributes them across the nodes in a round-robin fashion. However, it does not balance the incoming I/O requests on the Veritas Access cluster.

The following functionality is available in this feature:

- One of the existing Veritas Access virtual IP is configured as the load balancer IP.
- All clients can connect to the Veritas Access cluster using this single virtual IP.
- Veritas Access makes use of load balancer algorithms internally to allocate the next available Veritas Access node to serve the client.
Currently, the Veritas Access cluster makes use of the round-robin algorithm in the implementation of the load balancer.
- If the router node restarts, shuts down or halts for any reason, the IP load balancing is switched to another node automatically.
- If the serving node restarts, shuts down or halts for any reason, the node is removed automatically from the IP load balancing and the request served by that node is transferred to another node.

To configure the load balancer

- ◆ Enter the following command to configure the load balancer.

```
Network> loadbalance configure <virtual IP>
```

Note: The virtual IP that is to be configured as the IP load balancer should be added and brought online from CLISH before you execute the `loadbalance configure` command.

To view the status of the load balancer

- ◆ Enter the following command to view the status of the load balancer.

```
Network> loadbalance status
```

Note: The `Network> ip addr show` command also shows the virtual IP which acts as the IP load balancer in its output.

To remove the load balancer configuration

- ◆ Enter the following command to remove the load balancer configuration.

```
Network> loadbalance remove
```

To switch the load balancer virtual IP to another host manually

- ◆ Enter the following command to switch the load balancer virtual IP to another host manually.

```
Network> ip addr online load balancer virtual IP new node
```


Configuring authentication services

This chapter includes the following topics:

- [About configuring LDAP settings](#)
- [Configuring LDAP server settings](#)
- [Administering the Veritas Access cluster's LDAP client](#)
- [Configuring the NIS-related settings](#)
- [Configuring NSS lookup order](#)

About configuring LDAP settings

The Lightweight Directory Access Protocol (LDAP) is the protocol used to communicate with LDAP servers. The LDAP servers are the entities that perform the service. In Veritas Access, the most common use of LDAP is for user authentication.

For sites that use an LDAP server for access or authentication, Veritas Access provides a simple LDAP client configuration interface.

Before you configure Veritas Access LDAP settings, obtain the following LDAP configuration information from your system administrator:

- IP address or host name of the LDAP server. You also need the port number of the LDAP server.
- Base (or root) distinguished name (DN), for example:

```
cn=employees,c=us
```

LDAP database searches start here.

- Bind distinguished name (DN) and password, for example:

```
ou=engineering,c=us
```

This allows read access to portions of the LDAP database to search for information.

- Base DN for users, for example:

```
ou=users,dc=com
```

This allows access to the LDAP directory to search for and authenticate users.

- Base DN for groups, for example:

```
ou=groups,dc=com
```

This allows access to the LDAP database, to search for groups.

- Base DN for Netgroups, for example:

```
ou=netgroups,dc=com
```

This allows access to the LDAP database, to search for Netgroups.

- Root bind DN and password. This allows write access to the LDAP database, to modify information, such as changing a user's password.
- Secure Sockets Layer (SSL). Configures a cluster to use the Secure Sockets Layer (SSL) protocol to communicate with the LDAP server.
- Password hash algorithm, for example, `md5`, if a specific password encryption method is used with your LDAP server.

See [“Configuring LDAP server settings”](#) on page 58.

See [“Administering the Veritas Access cluster's LDAP client”](#) on page 61.

Configuring LDAP server settings

You can set the LDAP base Distinguished Name (base DN). LDAP records are structured in a hierarchical tree. You access records through a particular path, in this case, a Distinguished Name, or DN. The base DN indicates where in the LDAP directory hierarchy you want to start your search.

Note: For Veritas Access to access an LDAP directory service, you must specify the LDAP server DNS name or IP address.

To set the base DN for the LDAP server

- ◆ To set the base DN for the LDAP server, enter the following:

```
Network> ldap set basedn value
```

where *value* is the LDAP base DN in the following format:

```
dc=yourorg,dc=com
```

To set the LDAP server hostname or IP address

- ◆ To set the LDAP server hostname or IP address, enter the following:

```
Network> ldap set server value
```

where *value* is the LDAP server hostname or IP address.

To set the LDAP server port number

- ◆ To set the LDAP server port number, enter the following:

```
Network> ldap set port value
```

where *value* is the LDAP server port number.

To set Veritas Access to use LDAP over SSL

- ◆ To set Veritas Access to use LDAP over SSL, enter the following:

```
Network> ldap set ssl {on|off}
```

To set the bind DN for the LDAP server

- ◆ To set the bind DN for the LDAP server, enter the following:

```
Network> ldap set binddn value
```

where *value* is the LDAP bind DN in the following format:

```
cn=binduser,dc=yourorg,dc=com
```

The *value* setting is mandatory.

You are prompted to supply a password. You must use your LDAP server password.

To set the root bind DN for the LDAP server

- ◆ To set the root bind DN for the LDAP server, enter the following:

```
Network> ldap set rootbinddn value
```

where *value* is the LDAP root bind DN in the following format:

```
cn=admin,dc=yourorg,dc=com
```

You are prompted to supply a password. You must use your LDAP server password.

To set the LDAP users, groups, or netgroups base DN

- ◆ To set the LDAP users, groups, or netgroups base DN, enter the following:

```
Network> ldap set users-basedn value
```

```
Network> ldap set groups-basedn value
```

```
Network> ldap set netgroups-basedn value
```

users-basedn Specifies the value for the users-basedn. For example:
value ou=users,dc=example,dc=com (default)

groups-basedn Specifies the value for the groups-basedn. For example:
value ou=groups,dc=example,dc=com (default)

netgroups-basedn Specifies the value for the netgroups-basedn. For example:
value ou=netgroups,dc=example,dc=com (default)

To set the password hash algorithm

- ◆ To set the password hash algorithm, enter the following:

```
Network> ldap set password-hash {clear|crypt|md5}
```

To display the LDAP configured settings

- ◆ To display the LDAP configured settings, enter the following:

```
Network> ldap get {server|port|basedn|binddn|ssl|rootbinddn|  

users-basedn|groups-basedn|netgroups-basedn|password-hash}
```

To clear the LDAP settings

- ◆ To clear the previously configured LDAP settings, enter the following:

```
Network> ldap clear {server|port|basedn|binddn|ssl|rootbinddn|  
users-basedn|groups-basedn|netgroups-basedn|password-hash}
```

Administering the Veritas Access cluster's LDAP client

You can display the Lightweight Directory Access Protocol (LDAP) client configurations. LDAP clients use the LDAPv3 protocol to communicate with the server.

To display the LDAP client configuration

- ◆ To display the LDAP client configuration, enter the following:

```
Network> ldap show [users|groups|netgroups]
```

users	Displays the LDAP users that are available in the Name Service Switch (NSS) database.
groups	Displays the LDAP groups that are available in the NSS database.
netgroups	Displays the LDAP netgroups that are available in the NSS database.

If you do not include one of the optional variables, the command displays all the configured settings for the LDAP client.

To enable the LDAP client configuration

- ◆ To enable the LDAP client configuration, enter the following:

```
Network> ldap enable
```

LDAP clients use the LDAPv3 protocol for communicating with the server. Enabling the LDAP client configures the Pluggable Authentication Module (PAM) files to use LDAP. PAM is the standard authentication framework for Linux.

To disable the LDAP client configuration

- ◆ To disable the LDAP client configuration, enter the following:

```
Network> ldap disable
```

LDAP clients use the LDAPv3 protocol for communicating with the server. This command configures the PAM configuration files so that they do not use LDAP.

Configuring the NIS-related settings

Veritas Access supports Network Information Service (NIS), implemented in a NIS server, as an authentication authority. You can use NIS to authenticate computers.

If your environment uses NIS, enable the NIS-based authentication on the Veritas Access cluster.

Note: IPv6 addresses are not supported for NIS.

To display NIS-related settings

- ◆ To display NIS-related settings, enter the following:

```
Network> nis show [users|groups|netgroups]
```

users	Displays the NIS users that are available in the Veritas Access cluster's NIS database.
groups	Displays the NIS groups that are available in the Veritas Access cluster's NIS database.
netgroups	Displays the NIS netgroups that are available in the Veritas Access cluster's NIS database.

To set the NIS domain name on all nodes in the cluster

- ◆ To set the NIS domain name on the cluster nodes, enter the following:

```
Network> nis set domainname [domainname]
```

where *domainname* is the domain name.

To set NIS server name on all nodes in the cluster

- ◆ To set the NIS server name on all cluster nodes, enter the following:

```
Network> nis set servername servername
```

where *servername* is the NIS server name. You can use the server's name or IP address.

To enable NIS clients

- ◆ To enable NIS clients, enter the following:

```
Network> nis enable
```

To view the new settings, enter the following:

```
Network> nis show
```

To disable NIS clients

- ◆ To disable NIS clients, enter the following:

```
Network> nis disable
```

Configuring NSS lookup order

Name Service Switch (NSS) is a Veritas Access cluster service that provides a single configuration location to identify the services (such as NIS or LDAP) for network information such as hosts, groups, netgroups, passwords, and shadow files.

For example, host information may be on an NIS server. Group information may be in an LDAP database.

The NSS configuration specifies which network services the Veritas Access cluster should use to authenticate hosts, users, groups, and netgroups. The configuration also specifies the order in which multiple services should be queried.

To display the current value set on NSS for all groups, hosts, netgroups, passwd, and shadow files

- ◆ To display the current value set on nsswitch for all groups, hosts, netgroups, passwd, and shadow files

```
Network> nsswitch show
```

To change the order of group items

- ◆ To configure the NSS lookup order, enter the following:

```
Network> nsswitch conf {group|hosts|netgroups|passwd|shadow}  
value1 [[value2]] [[value3]] [[value4]]
```

group	Selects the group file.
hosts	Selects the hosts file.
netgroups	Selects the netgroups file.
passwd	Selects the password.
shadow	Selects the shadow file.
value	Specifies the following NSS lookup order with the following values: <ul style="list-style-type: none">■ <i>value1</i> (required) - { files/nis/winbind/ldap }■ <i>value2</i> (optional) - { files/nis/winbind/ldap }■ <i>value3</i> (optional) - { files/nis/winbind/ldap }■ <i>value4</i> (optional) - { files/nis/winbind/ldap }

For example:

```
Network> nsswitch conf group nis files  
Network> nsswitch show
```

To select DNS, you must use the following command:

```
Network> nsswitch conf hosts
```


Managing Veritas Access storage

- [Chapter 5. Configuring storage](#)
- [Chapter 6. Configuring data integrity with I/O fencing](#)
- [Chapter 7. Configuring iSCSI](#)
- [Chapter 8. Veritas Access as an iSCSI target](#)

Configuring storage

This chapter includes the following topics:

- [About storage provisioning and management](#)
- [About configuring disks](#)
- [About configuring storage pools](#)
- [Configuring storage pools](#)
- [About quotas for usage](#)
- [Enabling, disabling, and displaying the status of file system quotas](#)
- [Setting and displaying file system quotas](#)
- [Setting user quotas for users of specified groups](#)
- [About quotas for CIFS home directories](#)
- [Workflow for configuring and managing storage using the Veritas Access CLI](#)
- [Displaying information for all disk devices associated with the nodes in a cluster](#)
- [Displaying WWN information](#)
- [Importing new LUNs forcefully for new or existing pools](#)
- [Initiating host discovery of LUNs](#)
- [Increasing the storage capacity of a LUN](#)
- [Formatting or reinitializing a disk](#)
- [Removing a disk](#)

About storage provisioning and management

When you provision storage, you want to be able to assign the appropriate storage for the particular application. Veritas Access supports a variety of storage types. To help the users that provision the storage to select the appropriate storage, you classify the storage into groups called storage pools. A storage pool is a user-defined way to group the disks that have similar characteristics.

Veritas Access supports a wide variety of storage arrays, direct attached storage as well as in-server SSDs and HDDs. During the initial configuration, you add the disks to the Veritas Access nodes. For a storage array, a disk is a LUN from the storage array. For best performance and resiliency, each LUN should be provisioned to all Veritas Access nodes. Local disks and fully shared disks have unique names, but partially shared disks across nodes may have the same name. Make sure that you do not assign LUNs from the same enclosure to different nodes partially.

Before you can provision storage to Veritas Access, the physical LUNs must be set up and zoned for use with the Veritas Access cluster. The storage array administrator normally allocates and zones the physical storage.

Veritas Access does not support thin reclamation disks.

After the disks are correctly discovered by Veritas Access, you assign the disks to storage pools. You create a file system on one or more storage pools. You can mirror across different pools. You can also create tiers on different pools, and use SmartTier to manage file system data across those tiers.

You can also use local disks that are shared over the network. Both DAS disks and SAN disks (LUNs) can be used by the same cluster, and you can have a mix of DAS and SAN disks in the same storage pool.

About configuring disks

Disks and pools can be specified in the same command provided the disks are part of an existing storage pool.

The pool and disk that are specified first are allocated space before other pools and disks.

If the specified disk is larger than the space allocated, the remainder of the space is still utilized when another file system is created spanning the same disk.

About configuring storage pools

Veritas Access uses storage pools to provision storage. Pools are more a logical construct rather than an architectural component. Pools are loosely collections of disks.

In the Veritas Access context, a disk is a LUN provisioned from a storage array. Each LUN should be provisioned to all Veritas Access nodes. Disks must be added to pools prior to use.

During the initial configuration, you create storage pools, to discover disks, and to assign them to pools. Disk discovery and pool assignment are done once. Veritas Access propagates disk information to all the cluster nodes.

You must first create storage pools that can be used to build file systems on.

By default, all of the storage pools in Veritas Access share the same configuration. Copies of the configuration reside on disks in the storage pools. The first storage pool you create uses the default configuration. You can create additional storage pools to be part of that default configuration or to be isolated. An isolated storage pool protects the pool from losing the associated metadata even if all configuration disks in the main storage pool fail. If isolated storage pools exist, you cannot remove the disks in the non-isolated pool or destroy the last non-isolated pool.

Configuring storage pools

A storage pool is a group of disks that Veritas Access uses for allocation. Before creating a file system, you must create a storage pool.

To create the storage pool used to create a file system

- 1 List all of the available disks, and identify which ones you want to assign to which pools.

```
Storage> disk list
```

- 2 To create a storage pool, enter the following:

```
Storage> pool create pool_name disk1[,disk2,...] [isolated=yes|no]
```

pool_name	Specifies what the created storage pool will be named. The storage pool name should be a string.
disk1, disk2,...	Specifies the disks to include in the storage pool. If the specified disk does not exist, an error message is displayed. Use the <code>Storage> disk list</code> command to view the available disks. Each disk can only belong to one storage pool. If you try to add a disk that is already in use, an error message is displayed. To specify additional disks to be part of the storage pool, use a comma with no space in between.
isolated=yes no	Optional. Specifies whether or not the storage pool is isolated from other storage pools. Isolating the storage pool means that the configuration information is not shared. By default, storage pools are not isolated.

To add a set of disks to a logical pool

- ◆ To add a set of disks to a logical pool, enter the following:

```
Storage> pool adddisk pool_name disk1 [,disk2,...]
```

Where *pool_name* specifies the name of the storage pool to which the disks have to be added.

disk1, disk2,.. specifies the disks to be added to the pool.

To list your pools

- ◆ To list your pools, enter the following:

```
Storage> pool list
```

If a node is down, the `Storage> pool list` command shows local disks of that node.

To rename a pool

- ◆ To rename a pool, enter the following:

```
Storage> pool rename old_name new_name
```

<i>old_name</i>	Specifies the name for the existing pool that will be changed. If the old name is not the name of an existing pool, an error message is displayed.
<i>new_name</i>	Specifies the new name for the pool. If the specified new name for the pool is already being used by another pool, an error message is displayed.

To destroy a storage pool

- 1 Because you cannot destroy an unallocated storage pool, you need to remove the disk from the storage pool using the `Storage> pool rmdisk` command prior to trying to destroy the storage pool.

See [“About configuring disks”](#) on page 67.

If you want to move the disk from the unallocated pool to another existing pool, you can use the `Storage> pool mvdisk` command.

- 2 To destroy a storage pool, enter the following:

```
Storage> pool destroy pool_name
```

Where *pool_name* specifies the storage pool to delete. If the specified *pool_name* is not an existing storage pool, an error message is displayed.

If a node is down temporarily, it is not a good practice to destroy a storage pool that contains local disks of that node.

Note: You cannot destroy the last non-isolated pool if isolated pools exist.

To mark a disk as a spare disk

- ◆ To mark a disk as a spare disk and add it to a pool which can be later used for hot relocation, enter the following:

```
Storage> pool markdiskspare pool_name disk1 [, disk2,...]
```

Where *pool_name* specifies the name of the storage pool to which the disks have to be added.

disk1, disk2,.. specifies the disks to be marked as spare.

In case of failure of a disk or a plex, the affected sub disks are relocated to disks designated as spare disks.

To remove the spare disk flag set on a disk

- ◆ To remove the spare disk flag set on a disk by the `Storage> pool markdiskspare` command, enter the following:

```
Storage> pool removediskspare pool_name  
disk1 [, disk2,...]
```

Where *pool_name* specifies the name of the storage pool to which the disks belong.

disk1, disk2,.. specifies the disks from which the spare disk flag has to be removed.

To list free space for pools

- ◆ To list free space for your pool, enter the following:

```
Storage> pool free [pool_name]
```

Where *pool_name* specifies the pool for which you want to display free space information.

If a specified pool does not exist, an error message is displayed.

If *pool_name* is omitted, the free space for every pool is displayed, but information for specific disks is not displayed.

About quotas for usage

Disk quotas limit the usage for users or user groups. You can configure disk quotas for file systems or for CIFS home directories.

Note: Quotas work over NFS, but quota reporting and quota details are not visible over NFS.

Users and groups visible through different sources of name service lookup (nsswitch), local users, LDAP, NIS, and Windows users can be configured for file systems or CIFS home directory quotas.

There are two types of disk quotas:

- Usage quota (numspace) - limits the amount of disk space that can be used on a file system.
The numspace quota value must be an integer with a unit. The minimum unit is KB. VxFS calculates numspace quotas based on the number of KBs. The range for numspace is from 1K to 9007199254740991($2^{53} - 1$)K.
- Inode quota (numinodes) - limits the number of inodes that can be created on a file system.
An inode is a data structure in a UNIX or UNIX-like file system that describes the location of some or all of the disk blocks allocated to the file.
The numinodes quota value must be an integer without a unit, and the range is from 1 to 999999999999999999(19bit).
0 is valid for numspace and numinodes, which means the quota is infinite.

Veritas Access supports disk quota limits greater than 2 TB.

In addition to setting a limit on disk quotas, you can also define a warning level, or soft quota, whereby the Veritas Access administrator is informed that they are nearing their limit, which is less than the effective limit, or hard quota. Hard quota limits can be set so that a user is strictly not allowed to cross quota limits. A soft quota limit must be less than a hard quota limit for any type of quota.

Note: The alert for when a hard limit quota or a soft limit quota is reached in Veritas Access is not sent out immediately. The hard limit quota or soft limit quota alert is generated by a cron job scheduled to run daily at midnight.

Enabling, disabling, and displaying the status of file system quotas

To configure file system quotas, you must enable the file system quotas. You can enable file system quotas for all file systems or specify a file system name. You can enable quotas per user (**userquota**), quotas per group (**groupquota**), or both.

Once the quotas are enabled, you set the values for the number of blocks or the number of inodes that can be created. Quotas can be hard limits or soft limits.

See [“Setting and displaying file system quotas”](#) on page 73.

To enable a file system quota

- ◆ To enable a file system quota, enter the following:

```
Storage> quota fs enable [fs_name] [userquota | groupquota]
```

To disable a file system quota

- ◆ To disable a file system quota, enter the following:

```
Storage> quota fs disable [fs_name] [userquota | groupquota]
```

To display the status of a file system quota

- ◆ To display the status of a file system quota, enter the following:

```
Storage> quota fs status [fs_name] [userquota | groupquota]
```

Note: If the LDAP client is disabled, then the quota information may not be displayed using the `Storage> quota show` command for LDAP users and groups.

Setting and displaying file system quotas

You can set usage quotas for users or for groups.

Before the file system quotas take effect, you must enable the quotas for the file system.

See [“Enabling, disabling, and displaying the status of file system quotas”](#) on page 72.

To set the quota value

- ◆ To set the quota value for a file system, enter the following:

```
Storage> quota fs set {userquota | groupquota} user_or_group_names  
domain_name [hardlimit | softlimit] [numinodes | numspace]  
[value] [fs_name]
```

Domain name is the first section of the domain, for example:

```
veritas.example.com
```

The domain name is `veritas` in the example above.

If a value is not provided, the default value is used.

To set all quota values

- ◆ To set all of the quota values, enter the following:

```
Storage> quota fs setall {userquota | groupquota}  
[hardlimit | softlimit] [numinodes | numspace]  
[value] [fs_name]
```

For example, to set all existing user quotas to default values for the file system fs1:

```
Storage> quota fs show fs1  
  
Storage> quota fs setall userquota
```

To display the file system settings

- ◆ To display the file system settings, enter the following:

```
Storage> quota fs show [fs_name] [userquota | groupquota]  
[user_or_group_names]
```

For example, to display quota values for the file systemx:

```
Storage> quota fs show
```

Note: If the LDAP client is disabled, then the quota information may not be displayed using the `Storage> quota show` command.

fs_name	File system name you want to set the quota for.
userquota	User quota can be set with hard or soft limits on usage. Usage is dictated by the number of blocks and number of inodes that are created by the user.
groupquota	Group quota can be set with hard or soft limits on usage. Usage is dictated by the number of blocks and number of inodes that are created by all the users in the group.

<code>user_or_group_names</code>	<p>Name of the user or the name of the group for which a quota value is set.</p> <p>You can specify a comma-separated list of user or group names.</p> <p>To delete quota values for a user, you have to set all the user quota entries to 0. A user with a UID of 0 is not allowed in a <code>Storage> quota fs set</code> command.</p>
----------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

To set the default quota values

- ◆ To set the default quota values, enter the following:

```
Storage> quota fs setdefault {userquota | groupquota}
{hardlimit | softlimit} {numinodes | numspace} [value] [fs_name]
```

To display the default values

- ◆ To display the default values, enter the following:

```
Storage> quota fs showdefault [fs_name] [userquota | groupquota]
```

For example, to display the default quota values:

```
Storage> quota fs showdefault
```

Setting user quotas for users of specified groups

You can set the same quota for each user in a group with a single command. As with the other quota commands, you can specify hard or soft limits for the number of inodes or the usage space.

To set user quotas for users of specified groups

- ◆ To set user quotas on users of specified groups, enter the following:

```
Storage> quota fs setbygroup group_names domain_name
[hardlimit | softlimit] [numinodes | numspace]
[value] [fs_name]
```

About quotas for CIFS home directories

You use `Storage> quota cifshomedir` commands to configure quotas for CIFS home directories. Users and groups visible through different sources of name service

lookup (nsswitch), local users, LDAP, NIS, and Windows users can be configured for CIFS home directory quotas.

Default values are entered in a configuration file only. The actual application of the quota is done with the `set` and `setall` commands using the default values provided.

When a CIFS home directory file system is changed, quota information for a user's home directory is migrated from the existing home directory file system to the new home directory file system.

Quota migration results are based on the following logic:

- **Case 1:**

In the case where the existing home directory file system is NULL, you can set the new home directory file system to be multiple file systems (for example, fs1, fs2). If the multiple file systems previously had different quota values, the quota status and values from the first file system are migrated to other file systems in the new home directory. The first file system is the template. Only the user/group quota values that existed on the first file system are migrated. Other user/group quota values remain the same on the other file system.

For example, assume the following:

- The new home directory file systems are fs1 and fs2.
- user1, user2, and user3 have quota values on fs1.
- user2, user3, and user4 have quota values on fs2.

For the migration, user/group quota values for user1, user2, and user3 are migrated from fs1 to fs2. Quota values for user4 are kept the same on fs2, and user4 has no quota values on fs1.

- **Case 2:**

When the existing home directory file systems are already set, and you change the file systems for the home directory, the quota status and values need to be migrated from the existing home directory file systems to the new file systems. For this migration, the first file system in the existing home directory acts as the template for migrating quota status and values.

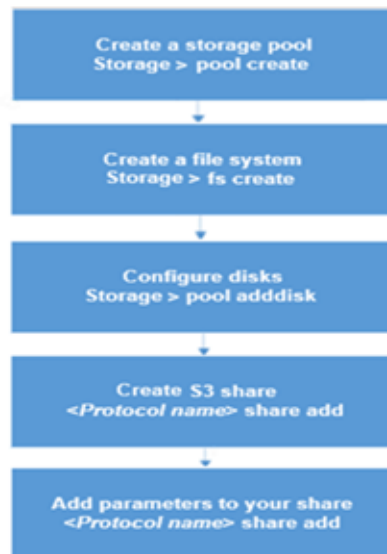
For example, if the existing home directory file systems are fs1 and fs2, and the file systems are changed to fs2, fs3, and fs4, then the user/group quota values on fs1 are migrated to fs3 and fs4. Other user/group values on fs3 and fs4 remain the same.

Workflow for configuring and managing storage using the Veritas Access CLI

Figure 5-1 describes configuring and managing storage using the Veritas Access CLI.

See the Veritas Access manual pages for the detailed syntax for completing the operations.

Figure 5-1 Workflow for configuring and managing Veritas Access storage using the CLI



Displaying information for all disk devices associated with the nodes in a cluster

You can display disk information for the disk devices associated with the nodes in the Veritas Access cluster. If local disks are present, the information includes entries for the local disks.

See the `storage_disk(1)` man page for the detailed examples.

The information displayed depends on the form of the command that you use. The following information is available:

Disk	Indicates the disk name.
Serial Number	Indicates the serial number for the disk.
Enclosure	Indicates the type of storage enclosure.
Size	Indicates the size of the disk.
Use%	Indicates the percentage of the disk that is being used.
Transport	Indicates transport protocol values like SCSI, FC, and other values.
ID	<p>ID column consists of the following four fields. A ":" separates these fields.</p> <ul style="list-style-type: none"> ■ VendorID - Specifies the name of the storage vendor, for example, HITACHI, IBM, EMC, and so on. ■ ProductID - Specifies the ProductID based on vendor. Each vendor manufactures different products. For example, HITACHI has HDS5700, HDS5800, and HDS9200 products. These products have ProductIDs such as DF350, DF400, and DF500. ■ TargetID - Specifies the TargetID. Each port of an array is a target. Two different arrays or two ports of the same array have different TargetIDs. TargetIDs start from 0. ■ LunID - Specifies the ID of the LUN. This should not be confused with the LUN serial number. LUN serial numbers uniquely identify a LUN in a target. Whereas a LunID uniquely identifies a LUN in an initiator group (or host group). Two LUNS in the same initiator group cannot have the same LunID. For example, if a LUN is assigned to two clusters, then the LunID of that LUN can be different in different clusters, but the serial number is the same.
Array Type	Indicates the type of storage array and can contain any one of the three values: Disk for JBODs, Active-Active, and Active-Passive.

To display a list of disks and nodes

- ◆ To display a list of disks and nodes, enter the following:

```
Storage> disk list
```

This form of the command displays local disk information for all nodes in the cluster.

To display the disk information

- ◆ To display the disk information, enter the following:

```
disk list detail
```

This form of the command displays local disk information from all the nodes in the cluster.

To display the disk list paths

- ◆ To display the disks multiple paths, enter the following:

```
Storage> disk list paths
```

This form of the command displays local disk information from all the nodes in the cluster.

Displaying WWN information

The `Storage> hba` (Host Bus Adapter) command displays World Wide Name (WWN) information for all of the nodes in the cluster. If you want to find the WWN information for a particular node, specify the node name (host name).

To display WWN information

- ◆ To display the WWN information, enter the following:

```
Storage> hba [host_name]
```

where you can use the `host_name` variable if you want to find WWN information for a particular node.

To display WWN information for all the running nodes in the cluster, enter the following:

To display WWN information for a particular node, enter the following:

```
Storage> hba [host_name]
```

HBA_Node_Name	Displays the node name for the Host Bus Adapter (HBA).
---------------	--------------------------------------------------------

WWN	Displays World Wide Name (WWN) information.
-----	---------------------------------------------

State	Available values include:
-------	---------------------------

- online
- offline

Speed	Displays the speed per second.
Support_Classes	Displays the class value from <code>/sys/class/fc_host/\${host}/supported_classes</code> .
Transmitted_FC_Frames	Displays a value equal to the number of total transmitted serial attached SCSI frames across all protocols.
Received_FC_frames	Displays a value equal to the number of total received serial attached SCSI frames across all protocols.
Link_Failure_Count	Displays a value equal to the value of the LINK FAILURE COUNT field of the Link Error Status.

Importing new LUNs forcefully for new or existing pools

The `Storage> scanbus force` command tries to import Logical Unit Numbers (LUNs) forcefully. This may help when using `Storage> scanbus` alone does not work.

To import LUNs forcefully

- ◆ To import LUNs forcefully, enter the following:

```
Storage> scanbus [force]
```

Initiating host discovery of LUNs

The `Storage> scanbus` command scans all of the SCSI devices connected to all of the nodes in the cluster. When you add new storage to your devices, you must scan for new SCSI devices. You only need to issue the command once and all of the nodes discover the newly added disks. The `scanbus` command updates the device configurations without interrupting the existing I/O activity. The scan does not inform you if there is a change in the storage configuration. You can see the latest storage configuration using the `Storage> disk list` command.

You do not need to reboot after `scanbus` has completed.

To scan SCSI devices

- ◆ To scan the SCSI devices connected to all of the nodes in the cluster, enter the following:

```
Storage> scanbus
```

Increasing the storage capacity of a LUN

The `Storage> disk grow` command lets you increase the storage capacity of a previously created LUN on a storage array disk.

Warning: When increasing the storage capacity of a disk, make sure that the storage array does not reformat it. This will destroy the data. For help, contact your Storage Administrator.

To increase the storage capacity of a LUN

- 1 Increase the storage capacity of the disk on your storage array. Contact your Storage Administrator for assistance.
- 2 Run the Veritas Access `Storage> scanbus` command to make sure that the disk is connected to the Veritas Access cluster.
See [“Initiating host discovery of LUNs ”](#) on page 80.
- 3 To increase the storage capacity of the LUN, enter the following:

```
Storage> disk grow disk_name
```

where *disk_name* is the name of the disk.

Formatting or reinitializing a disk

You can format or reinitialize a disk. If the disk does not belong to any group, the `Storage> disk format` command erases the first 100-MB space on the disk(s). You can format multiple disks at once.

If a DAS disk is formatted, it is exported to all the nodes in the cluster. DAS disks cannot be added to storage pools if they are not formatted.

To reformat or reinitialize a disk

- ◆ To reformat or reinitialize a disk, enter the following:

```
Storage> disk format disk1
```

Where *disk1* is the disk that you want to format or reinitialize.

Removing a disk

The `Storage> disk remove` command allows you to remove disks from a cluster. This command is helpful in situations when the disk attributes are incorrectly listed in Veritas Access.

Note: Only the disks that are not a part of a pool can be removed.

The `Storage> disk remove` command will not destroy the data on the disk, but it removes the disk from the system's configuration. Rebooting the cluster or running scanbus will bring back the disk into the system's configuration. To remove the disk permanently from the system's configuration, you should remove the disk's mapping from the array.

To remove a disk from a cluster

- ◆ To remove a disk from a cluster, enter the following:

```
Storage> disk remove disk1[,disk2,...]
```

disk1	Indicates the first disk name that you want to remove from the cluster.
-------	-------------------------------------------------------------------------

disk2	Indicates the second disk name that you want to remove from the cluster.
-------	--------------------------------------------------------------------------

Disk names are comma-separated without any spaces between the disk names.

Configuring data integrity with I/O fencing

This chapter includes the following topics:

- [About I/O fencing](#)
- [Configuring disk-based I/O fencing](#)

About I/O fencing

In the Veritas Access cluster, one method of communication between the nodes is conducted through heartbeats over private links. If the two nodes cannot communicate, the two nodes cannot verify each other's state. Neither node can distinguish if the failed communication is because of a failed link or a failed partner node. The network breaks into two networks that cannot communicate with each other but do communicate with the central storage. This condition is referred to as the "split-brain" condition.

I/O fencing protects data integrity if the split-brain condition occurs. I/O fencing determines which nodes retain access to the shared storage and which nodes are removed from the cluster, to prevent possible data corruption.

I/O fencing will be enabled on shared disks only if the disks are scsi-3 compliant.

In Veritas Access, I/O fencing has the following modes:

- Disk-based I/O fencing uses coordinator disks for arbitration in the event of a network partition. Coordinator disks are standard disks or LUNs that are set aside for use by the I/O fencing driver. All disks (both data and coordinator) must be scsi-3 compliant. The coordinator disks act as a global lock device during a cluster reconfiguration. This lock mechanism determines which node is allowed to fence off data drives from other nodes. A system must eject a peer from the coordinator disks before it can fence the peer from the data drives.

Racing for control of coordinator disks is how fencing helps prevent split-brain. Coordinator disks cannot be used for any other purpose. You cannot store data on them.

To use the disk-based I/O fencing feature, you enable fencing on each node in the cluster. Disk-based I/O fencing always requires an odd number of disks starting with three disks. You must also specify the three disks to use as coordinator disks. The minimum configuration must be a two-node cluster with Veritas Access software installed and more than three disks. Three of the disks are used as coordinator disks and the rest of the disks are used for storing data. See [“Configuring disk-based I/O fencing”](#) on page 84.

- Majority-based I/O fencing provides support for high availability when there are no additional servers or shared SCSI-3 disks that can act as coordination points. In case a split-brain condition occurs, the sub-cluster with more than half of the nodes remains online. If a sub-cluster has less than half of the nodes, then it panics itself. If the cluster has odd number of nodes, the sub-cluster which has the most number of cluster nodes survives and the sub-cluster with the least number of nodes is ejected out of the cluster. If the cluster has even number of nodes, the sub-cluster with the lowest cluster id survives.

For Veritas Access, majority-based fencing is used for Flexible Storage Sharing. Majority-based I/O fencing is administered only with the CLISH.

Configuring disk-based I/O fencing

To use the disk-based I/O fencing feature, the minimum configuration must be a two-node cluster with Veritas Access software installed and more than three disks. Three disks are used as coordinator disks and the rest of the disks are used for storing data.

Enabling I/O fencing configures disk-based fencing if shared disks are present.

Note: Enabling I/O fencing causes a disruption of Veritas Access services. When I/O fencing is enabled, some services such as NFS shares, CIFS shares, and S3 shares have to be stopped as the process of enabling fencing halts the VCS and cluster services. If any I/O is going on, then the cluster cannot be stopped. Veritas suggests that you bring down the Veritas Access services, enable I/O fencing, and then resume Veritas Access services.

To configure I/O fencing

- 1 To check the status of I/O fencing, enter the following:

```
Storage> fencing status
```

In the following example, I/O fencing is configured on the three disks *Disk_0*, *Disk_1* and *Disk_2* and the column header **Coord Flag On** indicates that these disks are in good condition. If you check the `Storage> disk list` output, it will be in the **OK** state.

- 2 If there are not three coordinator disks, you must add coordinator disks. You can add disks and enable fencing at the same time with the following command:

```
Storage> fencing on disk disk1,disk2,disk3
```

You may still provide three disks for fencing if three coordinator disks already exist. This will, however, remove the three coordinator disks previously used for fencing, and configure I/O fencing on the new disks.

Replacing an existing coordinator disk

You can replace a coordinator disk with another disk. The replacement disk must not be in a failed state, and must not be in use by an existing pool.

Note: If the disk being replaced is in a failed state, then you must delete the disk from the array. If the failed disk comes up and works properly, it can lead to an even number of fencing disks, and this affects the functionality.

To replace an existing coordinator disk

- ◆ To replace the existing coordinator disk, enter the following:

```
Storage> fencing replace src_disk  
dest_disk
```

where *src_disk* is the source disk and *dest_disk* is the destination disk.

Disabling I/O fencing

You can disable I/O fencing on all of the nodes. This operation does not free up the coordinator disks.

Note: Disabling I/O fencing causes a disruption of Veritas Access services. When I/O fencing is disabled, some services such as NFS shares, CIFS shares, and S3 shares have to be stopped as the process of disabling fencing halts the VCS and cluster services. If any I/O is going on, then the cluster cannot be stopped. Veritas suggests that you bring down the Veritas Access services, disable I/O fencing, and then resume Veritas Access services.

To disable I/O fencing

- ◆ To disable I/O fencing, enter the following:

```
Storage> fencing off
```

Destroying the coordinator pool

Destroys the coordinator pool if I/O fencing is disabled.

Note: This operation is not supported for a single-node cluster.

To destroy the coordinator pool

- ◆ To destroy the coordinator pool, enter the following:

```
Storage> fencing destroy
```

Configuring iSCSI

This chapter includes the following topics:

- [About iSCSI](#)
- [Configuring the iSCSI initiator](#)
- [Configuring the iSCSI initiator name](#)
- [Configuring the iSCSI devices](#)
- [Configuring discovery on iSCSI](#)
- [Configuring the iSCSI targets](#)
- [Modifying tunables for iSCSI](#)

About iSCSI

The Internet Small Computer System Interface (iSCSI) is an Internet protocol-based storage networking standard that links data storage facilities. By carrying SCSI commands over IP networks, iSCSI facilitates data transfers over Intranets and manages storage over long distances.

The iSCSI feature allows Veritas Access servers to use iSCSI disks as shared storage.

Configuring the iSCSI initiator

To display the iSCSI initiator service

- ◆ To display the status of the iSCSI initiator service, enter the following:

```
Storage> iscsi status
```

To start the iSCSI initiator service

- ◆ To start the iSCSI initiator service, enter the following:

```
Storage> iscsi start
```

To stop the iSCSI initiator service

- ◆ To stop the iSCSI initiator service, enter the following:

```
Storage> iscsi stop
```

Configuring the iSCSI initiator name

Veritas Access generates iSCSI initiator names for each node.

You can set the prefix that Veritas Access uses to generate initiator names. Veritas Access names each initiator with this prefix followed by the node number of the node.

To display the iSCSI initiator names

- ◆ To display the iSCSI initiator names, enter the following:

```
Storage> iscsi initiatorname list
```

To configure the iSCSI initiator name

- ◆ To configure the iSCSI initiator name prefix, enter the following:

```
Storage> iscsi initiatorname setprefix initiatorname-prefix
```

where *initiatorname-prefix* is a name that conforms to the naming rules for initiator and target names as specified in RFC3721. Initiator names for nodes in the cluster are generated by appending the node number to this prefix.

Configuring the iSCSI devices

The iSCSI initiator contains a list of network devices (network interfaces) from which connections are made to targets.

You can add or delete devices from this list.

When you add a device for use with the iSCSI initiator, iSCSI initiator connections use this device to connect to the target. If there are any existing targets, then the iSCSI initiator initiates a connection to all targets by using the newly set devices.

When you delete a *device* from the iSCSI configuration, any existing connections by way of the *device* to targets is terminated. If there are existing targets, you cannot delete the last device in the iSCSI initiator configuration.

To display the list of devices

- ◆ To display the list of devices, enter the following:

```
Storage> iscsi device list
```

To add an iSCSI device

- ◆ To add an iSCSI device, enter the following:

```
Storage> iscsi device add device
```

where *device* is the device where the operation takes place.

To delete an iSCSI device

- ◆ To delete an iSCSI device, enter the following:

```
Storage> iscsi device delete device
```

where *device* is the device where the operation takes place.

Configuring discovery on iSCSI

The iSCSI initiator contains a list of iSCSI target discovery addresses.

To display the iSCSI discovery addresses

- ◆ To display the iSCSI discovery addresses, enter the following:

```
Storage> iscsi discovery list
```

To add a discovery address to the iSCSI initiator

- 1 To add a discovery address to the iSCSI initiator, enter the following:

```
Storage> iscsi discovery add discovery-address
```

where:

discovery-address The target address at which an initiator can request a list of targets using a `SendTargets` text request as specified in iSCSI protocol of RFC3720.

You can specify either an IPv4 address or an IPv6 address. Optionally, you can specify a port with the IP address.

If no port is specified, the default port `3260` is used. Verify that your firewall allows you to access the target location through the port. For example:

```
# telnet discovery-address 3260
```

- 2 To verify the addition of the discovery address, display the discovery addresses.

```
Storage> iscsi discovery list
```

To delete an iSCSI discovery address

- 1 To delete the targets discovered using this discovery address, enter the following:

```
Storage> iscsi discovery del discovery-address
```

where:

discovery-address The target address at which an initiator can request a list of targets using a `SendTargets` text request as specified in iSCSI protocol of RFC3720.

You can specify either an IPv4 address or an IPv6 address. Optionally, you can specify a port with the IP address.

If no port is specified, the default port `3260` is used. Verify that your firewall allows you to access the target location through the port. For example:

```
# telnet discovery-address 3260
```

- 2 To verify the deletion of the discovery address, display the discovery addresses.

```
Storage> iscsi discovery list
```

To rediscover an iSCSI discovery address

- ◆ To rediscover an iSCSI discovery address, enter the following:

```
Storage> iscsi discovery rediscover discovery-address
```

where:

discovery-address The target address at which an initiator can request a list of targets using a `SendTargets` text request as specified in iSCSI protocol of RFC3720.

You can specify either an IPv4 address or an IPv6 address. Optionally, you can specify a port with the IP address.

If no port is specified, the default port `3260` is used. Verify that your firewall allows you to access the target location through the port. For example:

```
# telnet discovery-address 3260
```

To rediscover changes in targets or LUNs at a discovery address

- ◆ To rediscover changes in targets or LUNs at a discovery address, enter the following:

```
Storage> iscsi discovery rediscover_new discovery-address
```

where:

discovery-address The target address at which an initiator can request a list of targets using a `SendTargets` text request as specified in iSCSI protocol of RFC3720.

You can specify either an IPv4 address or an IPv6 address. Optionally, you can specify a port with the IP address.

If no port is specified, the default port 3260 is used. Verify that your firewall allows you to access the target location through the port. For example:

```
# telnet discovery-address 3260
```

New LUNs or targets discovered at *discovery-address* will be automatically added and logged into. This command does not discover any targets that have been deleted at *discovery-address*.

Configuring the iSCSI targets

To display the iSCSI targets

- ◆ To display the iSCSI targets, enter the following:

```
Storage> iscsi target list
```

To display the iSCSI target details

- ◆ To display the iSCSI target details, enter the following:

```
Storage> iscsi target listdetail target
```

where *target* is the name of the node you want to display the details for.

This list also shows targets discovered at *discovery-address*, not only manually added targets.

To add an iSCSI target

- ◆ To add an iSCSI target, enter the following:

```
Storage> iscsi target add target-name portal-address
```

target-name Name of the iSCSI target at which SCSI LUNs are available.
target-name should conform to the naming rules defined in RFC3721.

portal-address The location where the target is accessible.
 You can specify either an IPv4 address or an IPv6 address.

To delete an iSCSI target

- ◆ To delete an iSCSI target, enter the following:

```
Storage> iscsi target del target-name  

{discovery-address|portal-address}
```

target-name Name of the iSCSI target at which SCSI LUNs are available.
target-name should conform to the naming rules defined in RFC3721.

discovery-address Target address at which an initiator can request a list of targets using a `SendTargets` text request as specified in iSCSI protocol of RFC3720. If no port is specified with the discovery address, default port 3260 is used.

portal-address The location where the target is accessible.

To login to an iSCSI target

- ◆ To log in to an iSCSI target, enter the following:

```
Storage> iscsi target login target-name
{discovery-address | portal-address}
```

<i>target-name</i>	Name of the iSCSI target at which SCSI LUNs are available. <i>target-name</i> should conform to the naming rules defined in RFC3721.
<i>discovery-address</i>	Target address at which an initiator can request a list of targets using a <code>SendTargets</code> text request as specified in iSCSI protocol of RFC3720. If no port is specified with the discovery address, default port 3260 is used.
<i>portal-address</i>	The location where the target is accessible.

To logout from an iSCSI target

- ◆ To logout from an iSCSI target, enter the following:

```
Storage> iscsi target logout target-name
{discovery-address | portal-address}
```

<i>target-name</i>	Name of the iSCSI target at which SCSI LUNs are available. <i>target-name</i> should conform to the naming rules defined in RFC3721.
<i>discovery-address</i>	Target address at which an initiator can request a list of targets using a <code>SendTargets</code> text request as specified in iSCSI protocol of RFC3720. If no port is specified with the discovery address, default port 3260 is used.
<i>portal-address</i>	The location where the target is accessible.

To rescan targets for new LUNs

- ◆ To rescan a target for a new LUN, enter the following:

```
Storage> iscsi target rescan target-name
```

where *target-name* is the name of the iSCSI target that you want to rescan.

You can use the `Storage> iscsi target rescan` command for both static targets and discovered targets.

Modifying tunables for iSCSI

You can set the values of the attributes on the targets. You can set or show the default values, the values for all targets, or the values for a specific target.

[Table 7-1](#) shows the target attributes that you can modify.

Table 7-1 Attributes for iSCSI targets

Attribute	Description
<code>cmds_max</code>	The maximum number of SCSI commands that the session will queue. A session is defined as a connection between the initiator and target portal for accessing a given target. <code>cmds_max</code> defines the commands per target, which could be multiple LUNs. Valid values range from 2 to 2048 and should be a power of 2.
<code>fast_abort</code>	Defines whether initiator should respond to R2Ts (Request to Transfer) after sending a task management function like an <code>ABORT_TASK</code> or <code>LOGICAL UNIT RESET</code> . A value of Yes causes the initiator to stop responding to R2Ts after an <code>ABORT_TASK</code> request is received. For Equallogic arrays, the recommended value is No. Valid values are Yes or No.
<code>initial_login_retry_max</code>	The maximum number of times that the iSCSI initiator should try a login to the target during first login. This only affects the initial login. Valid values range from 1 to 16. During each login attempt, wait for <code>login_timeout</code> seconds for the login to succeed.
<code>login_timeout</code>	The amount of time that the iSCSI initiator service should wait for login to complete. The value of this attribute is in seconds. Valid values range from 10 to 600.
<code>logout_timeout</code>	The amount of time that the iSCSI initiator service should wait for logout to complete. The value of this attribute is in seconds. Valid values range from 10 to 600.
<code>noop_interval</code>	The time to wait between subsequent sending of Nop-out requests. The value of this attribute is in seconds. Valid values range from 5 to 600.

Table 7-1 Attributes for iSCSI targets (*continued*)

Attribute	Description
<code>noop_timeout</code>	The amount of time that the iSCSI initiator service should wait for response to a Nop-out request sent to the target, before failing the connection. Failing the connection causes the I/O to be failed and retried on any other available path. The value of this attribute is in seconds. Valid values range from 5 to 600.
<code>queue_depth</code>	The maximum number of SCSI commands queued per LUN, belonging to a target. The value for <code>queue_depth</code> cannot be greater than <code>cmds_max</code> . Valid values range from 1 to 128.
<code>replacement_timeout</code>	The amount of time to wait for session re-establishment before failing SCSI commands. The value of this attribute is in seconds. Valid values range from 10 to 86400.

To display the default value for target attributes

- ◆ To display the default value for target attributes, enter the following:

```
Storage> iscsi target attr showdefault
```

To display values for target attributes of all known targets

- ◆ To display values for target attributes of all known targets, enter the following:

```
Storage> iscsi target attr showall
```

To display the attribute values for a specific target

- ◆ To display the attribute values for a specific target, enter the following:

```
Storage> iscsi target attr show target-name
```

where *target-name* is the name of the iSCSI target to be displayed.

To set the default value for a target attribute

- ◆ To set the default value for a target attribute, enter the following:

```
Storage> iscsi target attr setdefault attribute value
```

attribute The attribute for which to set the value.

value The default value to be set for the attribute.

The default value is inherited by any new targets that get added.

To set an attribute value for all known targets

- ◆ To set an attribute value for all known targets, enter the following:

```
Storage> iscsi target attr setall attribute value
```

attribute The attribute for which to set the value.

value The value to be set for the attribute.

This command does not change the default value as shown in the `Storage> iscsi target attr showdefault` command. Changes to values are effective after re-login.

To set the attribute value for a specific target

- ◆ To set the attribute value for a specific target, enter the following:

```
Storage> iscsi target attr set target-name attribute value
```

target-name The name of the specific iSCSI target.

attribute The attribute of the specific target.

value The value to be set for the target attribute.

Veritas Access as an iSCSI target

This chapter includes the following topics:

- [About Veritas Access as an iSCSI target](#)
- [Managing the iSCSI targets](#)
- [Managing the iSCSI target service](#)
- [Managing the LUNs](#)
- [Managing the mappings with iSCSI initiators](#)
- [Managing the users](#)

About Veritas Access as an iSCSI target

Veritas Access as an iSCSI target is introduced for RHEL 7. 3 and 7.4 in this release. This feature enables a Veritas Access cluster to serve block storage. Through the use of multiple portal IPs, an iSCSI target can be served in active-active fashion.

This enables the block storage to be capable of supporting multi-pathing at the initiator end. Veritas Access eases provisioning of block storage with the functionality to resize, clone, and snapshot the LUNs, ACL controls such as initiator mapping and user management.

Note: Veritas Access as an iSCSI target supports VMware version 5.5.0 as an initiator.

You can perform the following functions on an iSCSI target:

- Start, stop, and check status of an iSCSI target service
- Create, destroy, check status, and list iSCSI targets
- Add and delete multiple portal addresses
- Add, delete, resize, manage, grow, shrink LUNs, and clone LUNs snapshots
- Map and remove mapping of iSCSI initiators
- Add and delete users to set up CHAP authentication
- Support for multiple portal IPs per target makes the targets active-active

Note: Veritas Access as an iSCSI target is a technical preview feature for RHEL 6.x in this release.

Managing the iSCSI targets

To create an iSCSI targets

- ◆ To create an iSCSI targets, enter the following:

```
Target> iscsi target create target-name
```

target-name	Name of the iSCSI target at which SCSI LUNs are available. <code>target-name</code> should conform to the naming rules defined in RFC3721 and it must be in lowercase.
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------

To destroy an iSCSI target

- ◆ To destroy an iSCSI target, enter the following:

```
Target> iscsi target destroy target-name
```

To list all iSCSI targets

- ◆ To list all iSCSI targets, enter the following:

```
Target> iscsi target list
```

To list an iSCSI target

- ◆ To list an iSCSI target, enter the following:

```
Target> iscsi target list target-name
```

To check the status of a specific iSCSI target

- ◆ To check the status of a specific iSCSI target, enter the following:

```
Target> iscsi target status target-name
```

To add multiple portal addresses to an iSCSI target

- ◆ To add multiple portal addresses to an iSCSI target, enter the following:

```
Target> iscsi target portal add target-name portal-ip
```

target-name	Name of the iSCSI target at which SCSI LUNs are available. target-name should conform to the naming rules defined in RFC3721 and it must be in lowercase.
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

portal-ip	The location where the target is accessible. You can specify multiple portal addresses that you want to add.
-----------	--------------------------------------------------------------------------------------------------------------

To delete multiple portal addresses from an iSCSI target

- ◆ To delete multiple portal addresses from an iSCSI target, enter the following:

```
Target> iscsi target portal del target-name portal-ip
```

target-name	Name of the iSCSI target at which SCSI LUNs are available. target-name should conform to the naming rules defined in RFC3721 and it must be in lowercase.
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

portal-ip	The location where the target is accessible. You can specify multiple portal addresses that you want to delete.
-----------	-----------------------------------------------------------------------------------------------------------------

To map a file system with a specified iSCSI target

- ◆ To map a file system with a specified iSCSI target, enter the following:

```
Target> iscsi target store add fs-name target-name
```

fs-name	Name of the file system which is to be mapped to an iSCSI target.
---------	-------------------------------------------------------------------

target-name	Name of the iSCSI target at which SCSI LUNs are available. target-name should conform to the naming rules defined in RFC3721 and it must be in lowercase.
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

To remove the file system from a specified iSCSI target

- ◆ To remove the file system from a specified iSCSI target, enter the following:

```
Target> iscsi target store delete fs-name target-name
```

fs-name	Name of the file system which is to be removed from an iSCSI target.
---------	----------------------------------------------------------------------

target-name	Name of the iSCSI target at which SCSI LUNs are available. target-name should conform to the naming rules defined in RFC3721 and it must be in lowercase.
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

Managing the iSCSI target service

To start the iSCSI target service

- ◆ To start the iSCSI target service, enter the following:

```
Target> iscsi service start
```

To stop the iSCSI target service

- ◆ To stop the iSCSI target service, enter the following:

```
Target> iscsi service stop
```

To check the status of the iSCSI target service

- ◆ To check the status of the iSCSI target service, enter the following:

```
Target> iscsi service status
```

Managing the LUNs

To create a LUN with the specified name and size

- ◆ To create a LUN with the specified name and size on the specified file system, enter the following:

```
Target> iscsi lun create lun-name target-name size option=dense|sparse
```

lun-name	Name of the LUN which is to be created on the store that is associated with an iSCSI target, with specific name and size.
target-name	Name of the iSCSI target at which SCSI LUNs are available. <code>target-name</code> should conform to the naming rules defined in RFC3721 and it must be in lowercase.
size	Size of the LUN that you want to create.
option	Name of the LUN type that is to be specified, such as Dense or Sparse. By default, Sparse is selected.

To destroy a specific LUN

- ◆ To destroy a specific LUN, enter the following:

```
Target> iscsi lun destroy [lun-name | clone-name]
```

lun-name	Name of the LUN which is to be destroyed from an iSCSI target.
clone-name	Name of the iSCSI target at which SCSI LUNs are available. <code>clone-name</code> should conform to the naming rules defined in RFC3721 and it must be in lowercase.

To list the details of all the LUNs and clones present in all targets

- ◆ To list the details of all the LUNs and clones present in all targets, enter the following:

```
Target> iscsi lun list
```

To list the details of all the LUNs and clones present in a specific target

- ◆ To list the details of all the LUNs and clones present in a specific target, enter the following:

```
Target> iscsi lun list target-name
```

To grow LUN size to specified size

- ◆ To grow LUN size to specified size, enter the following:

```
Target> target iscsi lun growto lun-name target-name size
```

lun-name	Name of the LUN for which size is to be increased on the store that is associated with an iSCSI target.
target-name	Name of the iSCSI target at which SCSI LUNs are available. <code>target-name</code> should conform to the naming rules defined in RFC3721 and it must be in lowercase.
size	A new size is to be specified for the LUN.

To shrink LUN size to specified size

- ◆ To shrink LUN size to specified size, enter the following:

```
Target> target iscsi lun shrinkto lun-name target-name size
```

lun-name	Name of the LUN for which, size is to be decreased on the store that is associated with an iSCSI target.
target-name	Name of the iSCSI target at which SCSI LUNs are available. <code>target-name</code> should conform to the naming rules defined in RFC3721 and it must be in lowercase.
size	A new size is to be specified for the LUN.

To create a clone of a specified LUN

- ◆ To create a clone of LUN, enter the following:

```
Target> iscsi lun clone create lun-name clone-name
```

lun-name	Name of the LUN for which a clone is to be created on the store that is associated with an iSCSI target.
clone-name	Name of the clone of a LUN which is to be specified.

To list details of all clones

- ◆ To list details of all clones, enter the following:

```
Target> iscsi lun clone list
```

To list details of the specified clone

- ◆ To list details of the specified clone, enter the following:

```
Target> iscsi lun clone list clone-name
```

To create a snapshot of LUN

- ◆ To create a snapshot of LUN, enter the following:

```
Target> iscsi lun snapshot create lun-name snapshot-name
```

lun-name Name of the LUN for which a snapshot is to be created on the store that is associated with an iSCSI target.

snapshot-name Name of the snapshot of a LUN which is to be specified.

To destroy a snapshot of LUN

- ◆ To destroy a snapshot of LUN, enter the following:

```
Target> iscsi lun snapshot destroy lun-name snapshot-name
```

lun-name Name of the LUN for which a snapshot is to be destroyed from an iSCSI target.

snapshot-name Name of the snapshot of a LUN which is to be specified.

To restore a specified snapshot of specified LUN

- ◆ To restore a specified snapshot of specified LUN, enter the following:

```
Target> iscsi lun snapshot restore lun-name snapshot-name
```

lun-name Name of the LUN for which a snapshot is to be restored on the store that is associated with an iSCSI target.

snapshot-name Name of the snapshot of a LUN which is to be specified.

To list details of all the snapshots

- ◆ To list details of all the snapshots, enter the following:

```
Target> iscsi lun snapshot list
```


To list details of a specified snapshot

- ◆ To list details of a specified snapshot, enter the following:

```
Target> iscsi lun snapshot list snapshot-name
```

where *snapshot-name* is the snapshot name for which details are displayed.

Managing the mappings with iSCSI initiators

To map an iSCSI initiator to a specific iSCSI target

- ◆ To map an iSCSI initiator to a specific iSCSI target, enter the following:

```
Target> iscsi target map add target-name initiator-name
```

target-name Name of the iSCSI target where the SCSI LUNs are available.
 target-name should conform to the naming rules defined in RFC3721 and it must be in lowercase.

initiator-name where *initiator-name* is a name that conforms to the naming rules for initiator and target names as specified in RFC3721 and it must be in lowercase. Initiator names for nodes in the cluster are generated by appending the node number.

To remove the mapping of an iSCSI initiator from specific iSCSI target

- ◆ To remove the mapping of iSCSI initiator from specific iSCSI target, enter the following:

```
Target> iscsi target map delete target-name initiator-name
```

target-name Name of the iSCSI target at which SCSI LUNs are available.
 target-name should conform to the naming rules defined in RFC3721 and it must be in lowercase.

initiator-name where *initiator-name* is a name that conforms to the naming rules for initiator and target names as specified in RFC3721 and it must be in lowercase. Initiator names for nodes in the cluster are generated by appending the node number.

Managing the users

To create an incoming user and bind the account to a specified, existing iSCSI target

- ◆ To create an incoming user and bind the account to a specified, existing iSCSI target, enter the following:

```
Target> iscsi target auth incominguser add target-name user-name
```

target-name Name of the iSCSI target at which SCSI LUNs are available.
 target-name should conform to the naming rules defined in RFC3721 and it must be in lowercase.

user-name Name of the incoming user that is to be added to an iSCSI target.

To remove an incoming user and unbind the account from its corresponding iSCSI target

- ◆ To remove an incoming user and unbind the account from its corresponding iSCSI target, enter the following:

```
Target> iscsi target auth incominguser delete target-name user-name
```

target-name Name of the iSCSI target at which SCSI LUNs are available.
 target-name should conform to the naming rules defined in RFC3721 and it must be in lowercase.

user-name Name of the incoming user that is to be removed from an iSCSI target.

Managing Veritas Access file access services

- [Chapter 9. Configuring your NFS server](#)
- [Chapter 10. Using Veritas Access as a CIFS server](#)
- [Chapter 11. Configuring Veritas Access to work with Oracle Direct NFS](#)
- [Chapter 12. Configuring an FTP server](#)

Configuring your NFS server

This chapter includes the following topics:

- [About using NFS server with Veritas Access](#)
- [Using the kernel-based NFS server](#)
- [Using the NFS-Ganesha server](#)
- [Switching between NFS servers](#)
- [Recommended tuning for NFS-Ganesha version 3 and version 4](#)
- [Accessing the NFS server](#)
- [Displaying and resetting NFS statistics](#)
- [Configuring Veritas Access for ID mapping for NFS version 4](#)
- [Configuring the NFS client for ID mapping for NFS version 4](#)
- [About authenticating NFS clients](#)
- [Setting up Kerberos authentication for NFS clients](#)

About using NFS server with Veritas Access

Veritas Access provides file access services to UNIX and Linux client computers using the Network File System (NFS) protocol. Veritas Access supports NFSv3 and NFSv4. Veritas Access provides the following NFS server support:

- Kernel-based NFS server
See [“Using the kernel-based NFS server”](#) on page 109.

- NFS-Ganesha server

See [“Using the NFS-Ganesha server”](#) on page 109.

At any time, either NFS-Ganesha or kernel NFS is active. The kernel NFS server is enabled by default. If required, you can switch the NFS server that you use.

Use the NFS-Ganesha server if you are using a scale-out file system.

See [“Switching between NFS servers”](#) on page 109.

Using the kernel-based NFS server

The kernel-based NFS server supports NFS version 3 and version 4. The kernel NFS server is enabled by default. Kernel NFS supports Active-Active mode serving NFS version 3 and 4. Veritas recommends that you use the default kernel-based NFS server.

Using the NFS-Ganesha server

If you plan to use a scale-out filesystem with 'largefs' layout, you must use Veritas Access with an NFS-Ganesha server.

NFS-Ganesha provides support for both NFS version 3 and NFS version 4. NFS-Ganesha is a user-space implementation of the NFS server. The use of a NFS-Ganesha server is optional. NFS-Ganesha is not enabled by default.

Certain limitations apply for NFS-Ganesha.

See the *Veritas Access Release notes* for the limitations.

Since the kernel-based NFS server is the default, switch the NFS server to NFS-Ganesha.

Switching between NFS servers

If NFS v3 or NFS v4 is your primary use case, then we recommend that you use the kernel NFS server. Both NFS v3 and NFS v4 support Kerberos authentication.

The NFS-Ganesha server supports both NFS v3 and NFS v4.

A CLISH command is provided to switch from kernel NFS server to NFS-Ganesha, or vice versa. Before you switch between the NFS servers, the NFS server must be offline.

All of the available NFS shares are moved from the previous NFS server to the new NFS server; therefore, the operation may be time consuming.

When the NFS server is switched from kernel-based NFS to NFS-Ganesha or vice versa, the existing NFS mounts on the client are no longer active. The client is required to remount the exports to access the shares.

To switch between NFS servers

- 1 Make sure that the NFS server is offline. You can view the status of the NFS server with the following command:

```
NFS> server status
```

- 2 Use the following command to switch the NFS server:

```
NFS> server switch
```

Recommended tuning for NFS-Ganesha version 3 and version 4

Veritas Access supports both the NFS kernel-based server and the NFS-Ganesha server in a mutually exclusive way. The NFS kernel-based server supports NFS version 3 and version 4. The NFS-Ganesha server also supports both NFS version 3 and NFS version 4.

See [“Using the NFS-Ganesha server”](#) on page 109.

The NFS-Ganesha server does not run in the kernel, instead NFS-Ganesha runs in user space on the NFS server. This means that the NFS-Ganesha server processes can be affected by system resource limitations as any other user space process can be affected. There are some NFS-server operating system tuning values that you should modify to ensure that the NFS-Ganesha server performance is not unduly affected. You use the NFS client mount option `version` to determine whether NFS version 3 or NFS version 4 is used. On the NFS client, you can select either the `version=3` or the `version=4` mount option. The NFS client is unaware of whether the NFS server is using kernel-based NFS or NFS-Ganesha.

When you start a system, `kswapd_init()` calls a kernel thread that is called `kswapd`, which continuously executes the function `kswapd()` in `mm/vmscan.c` that usually sleeps. The `kswapd` daemon is responsible for reclaiming pages when memory is running low. `kswapd` performs most of the tasks that are needed to maintain the page cache correctly, shrink slab caches, and swap out processes if necessary. `kswapd` keeps freeing pages until the `pages_high` watermark is reached. Under extreme memory pressure, processes do the work of `kswapd` synchronously by calling `balance_classzone()`, which calls the `try_to_free_pages_zone()`.

When there is memory pressure, pages are claimed using two different methods.

- `pgscank/s` – The `kswapd` kernel daemon periodically wakes up and claims (frees) memory in the background when free memory is low. `pgscank/s` records this activity.
- `pgscand/s` – When `kswapd` fails to free up enough memory, then the memory is also claimed directly in the process context (thus blocking the user program execution). `pgscand/s` records this activity.
- The total pages being claimed (also known as page stealing) is therefore a combination of both `pgscank/s` and `pgscand/s`. `pgsteal/s` records the total activity, so $(pgsteal/s = pgscank/s + pgscand/s)$.

The NFS-Ganesha user process can be affected when `kswapd` fails to free up enough memory. To alleviate the possibility of the NFS-Ganesha process from doing the work of `kswapd`, Veritas recommends increasing the value of the Linux virtual machine tunable `min_free_kbytes`.

Example of a default auto-tuned value:

```
sysctl -a | grep vm.min_free
vm.min_free_kbytes = 90112
```

You use `min_free_kbytes` to force the Linux VM (virtual memory management) to keep a minimum number of kilobytes free. The VM uses this number to compute a watermark value for each `lowmem` zone in the system.

Table 9-1 Recommended tuning parameters for NFS version 3 and version 4

Option	Description
NFS mount options	File system mount options for the NFS client: <ul style="list-style-type: none"> ■ <code>version=3/4</code> ■ <code>nordirplus</code> ■ <code>sharecache</code>
NFS server export options	NFS server export options: <ul style="list-style-type: none"> ■ <code>rw</code> ■ <code>sync</code> ■ <code>no_root_squash</code>

Table 9-1 Recommended tuning parameters for NFS version 3 and version 4 (*continued*)

Option	Description
Jumbo frames	A jumbo frame is an Ethernet frame with a payload greater than the standard maximum transmission unit (MTU) of 1,500 bytes. Enabling jumbo frames improves network performance in I/O intensive workloads. If jumbo frames are supported by your network, and if you wish to use jumbo frames, Veritas recommends using a jumbo frame size of 5000.
<code>min_free_kbytes</code>	On server nodes with 96 GB RAM or more, the recommended value of <code>min_free_kbytes</code> is 1048576 (=1 GB). On server nodes using the minimum of 32 GB RAM, the minimum recommended value of <code>min_free_kbytes</code> is 524288 (=512 MB).

Accessing the NFS server

To check on the NFS server status

- ◆ Prior to starting the NFS server, check on the status of the server by entering:

```
NFS> server status
```

The output shows the status. The output also indicates whether the NFS server used is the kernel NFS server or the NFS-Ganesha server.

The states (ONLINE, OFFLINE, and FAULTED) correspond to each Veritas Access node identified by the node name. The states of the node may vary depending on the situation for that particular node.

The possible states of the `NFS> server status` command are:

ONLINE	Indicates that the node can serve NFS protocols to the client.
OFFLINE	Indicates the NFS services on that node are down.
FAULTED	Indicates something is wrong with the NFS service on the node.

You can run the `NFS> server start` command to restart the NFS services, and only the nodes where NFS services have problems, are restarted.

To start the NFS server

- ◆ To start the NFS server, enter the following:

```
NFS> server start
```

You can use the `NFS> server start` command to clear an OFFLINE state from the `NFS> server status` output by only restarting the services that are offline. You can run the `NFS> server start` command multiple times without it affecting the already-started NFS server.

Run the `NFS> server status` command again to confirm the change.

To stop the NFS server

- ◆ To stop the NFS server, enter the following:

```
NFS> server stop
```

Displaying and resetting NFS statistics

The NFS statistics shown differs depending on whether the NFS server is the default kernel NFS server, or the NFS-Ganesha server.

Veritas Access does not support resetting the NFS statistics for the NFS-Ganesha server.

To display statistics for a specific node or for all the nodes in the cluster

To display NFS statistics, enter the following:

```
NFS> stat show [nodename]
```

where *nodename* specifies the node name for which you are trying to obtain the statistical information. If the *nodename* is not specified, statistics for all the nodes in the cluster are displayed.

To display the NFS statistics for all the nodes in the cluster for the kernel NFS server, enter the following:

```
NFS> stat show all
```

To display the NFS statistics for all the nodes in the cluster for the NFS-Ganesha server, enter the following:

```
NFS> stat show all
```

To reset NFS statistics for a specific node or for all the nodes in the cluster to zero

- ◆ To reset NFS statistics for the kernel NFS server, enter the following:

```
NFS> stat reset [nodename]
```

where *nodename* specifies the node name for which you want to reset the NFS statistics to zero. If *nodename* is not specified, NFS statistics for all the nodes in the cluster are reset to zero. Statistics are automatically reset to zero after a reboot of a node, or in the case of NFS-Ganesha, after you reboot the node or the NFS server restarts.

Configuring Veritas Access for ID mapping for NFS version 4

If you plan to use NFS version 4, you must configure Veritas Access to map the user IDs to the required format. In NFS version 3, each user is identified by a number, the user ID (uid). A UNIX file also identifies the owner of the file by a uid number. NFS version 4 has a different way of identifying users than that used by NFS version 3. In NFS version 4, each user is identified by a string, such as `user1@example.com`.

Veritas Access requires a mechanism to map the user strings from NFS version 4 to uids on the server and the client. This process, called ID mapping, uses a file `/etc/idmapd.conf`.

NFS version 4 uses the `/etc/idmapd.conf` file to map the IDs. The Domain field needs to be set to the DNS domain of the Veritas Access server. If the DNS domain is not set, the ID mapping maps all of the users on the client to the user 'nobody'.

To configure Veritas Access for ID mapping

- ◆ Configure the DNS domain of Veritas Access using the following command:

```
Network> dns set domainname domainname
```

When the NFS server is started, the `/etc/idmapd.conf` file is updated with the domain information of the Veritas Access server.

You must also configure the NFS client.

Configuring the NFS client for ID mapping for NFS version 4

For NFS version 4, you must configure the NFS client so that the NFS version 4 user strings can be mapped to the uids. You must also configure the NFS server.

To configure the NFS client for ID mapping

- 1 For proper ID mapping, set the `Domain` field in the `/etc/ldapd.conf` file as the DNS domain name of the NFS client. Make sure that the DNS domain is the same for the NFS client and the Veritas Access server.

This setting in the `/etc/ldapd.conf` file should be updated on the NFS client.

- 2 Clear the ID mapping cache on the NFS client using the command `nfsidmap -c` and restart the ID mapping service.

About authenticating NFS clients

See [“About managing NFS shares using netgroups”](#) on page 281.

Both the NFS-Ganesha server and kernel NFS server support Kerberos authentication.

Setting up Kerberos authentication for NFS clients

Kerberos provides a secure way of authenticating NFS clients. In this configuration, the Veritas Access server behaves as a Kerberos client. The Kerberos KDC (Key Distribution Center) server must already be set up and running outside of Veritas Access. For NFS version 3, when a Veritas Access share is exported with the `krb5` security option, the NFS clients have to mount the Veritas Access share with the `krb5` mount option. Otherwise the mount fails with an authentication error. For NFS version 4, the NFS clients automatically find the security type and mount the Veritas Access share with the same mount option.

Note: When CIFS security is configured with `ads`, Kerberos for NFS cannot be configured. When NFS is configured for Kerberos authentication, CIFS security cannot be configured with `ads`.

To configure Veritas Access for authenticating NFS clients using Kerberos, perform the tasks in the order that is listed in [Table 9-2](#).

Table 9-2 Tasks for configuring Veritas Access for authenticating NFS clients using Kerberos

Task	Where to find more information
Add and configure Veritas Access to the Kerberos realm	See “Adding and configuring Veritas Access to the Kerberos realm” on page 116.
Configure the NFS server for ID mapping	See “Configuring Veritas Access for ID mapping for NFS version 4” on page 114.
Configure the NFS client for ID mapping	See “Configuring the NFS client for ID mapping for NFS version 4” on page 115.
Exporting an NFS share for Kerberos authentication	See “Exporting an NFS share for Kerberos authentication” on page 283.
Mount the NFS share from the NFS client	See “Mounting an NFS share with Kerberos security from the NFS client” on page 284.

Adding and configuring Veritas Access to the Kerberos realm

Kerberos authentication support on Veritas Access is available only if the Key Distribution Center (KDC) server is running on a standalone computer (in a non-AD (Active Directory) environment), and there is a single KDC server. Before Veritas Access can be used as a Kerberos client, the NFS service principal of Veritas Access has to be added to the KDC server. Use the Veritas Access cluster name (either the short name or the fully qualified domain name) in small letters as the host name when creating the NFS service principal.

For example, if `access_ga_01` and `access_ga_02` are two nodes in the Veritas Access cluster, then `access_ga` (or the fully qualified domain name `access_ga.example.com`) should be used for adding the NFS service principal. The Domain Name System (DNS) or `/etc/hosts` is then set up to resolve `access_ga` to all the virtual IPs of the Veritas Access cluster.

To configure the KDC server

- 1 Create the NFS service principal on the KDC server using the `kadmin.local` command.

```
addprinc -randkey nfs/access_ga
```

- 2 Create a `keytab` file for the NFS service principal on KDC.

```
ktadd -k /etc/access.keytab nfs/access_ga
```

- 3 Copy the created `keytab` file (`/etc/access.keytab`) to the Veritas Access console node.

- 4 Use the `Network> krb standalone set` command to set the Kerberos configuration on Veritas Access.

The `Network> krb standalone set` command takes the KDC server name, Kerberos realm, and the location of the `keytab` that is located on the Veritas Access console node. This command sets up the Kerberos configuration file `/etc/krb5.conf` with the KDC server name and realm on all the nodes of the Veritas Access cluster. The command then copies the `keytab` file to `/etc/krb5.keytab` on all the nodes of the Veritas Access cluster.

```
Network> krb standalone set kdc_server TESTKDC.COM  
/home/support/krb5.keytab
```

The `Network> krb standalone set` command checks for the correct domain in the `/etc/idmapd.conf` file. If the domain is not set, the command gives a warning message saying that the DNS domain name needs to be set.

See [“Configuring Veritas Access for ID mapping for NFS version 4”](#) on page 114.

- 5 Use the `Network> krb standalone show` command to show the Kerberos configuration.
- 6 Use the following commands to stop and restart the NFS-Ganesha service:

```
NFS> server stop  
NFS> server start
```

- 7 Use the `Network> krb standalone unset` command to reset the Kerberos configuration.

After the KDC server is configured, you can export the NFS shares with Kerberos authentication options.

Using Veritas Access as a CIFS server

This chapter includes the following topics:

- [About configuring Veritas Access for CIFS](#)
- [About configuring CIFS for standalone mode](#)
- [Configuring CIFS server status for standalone mode](#)
- [Changing security settings](#)
- [Changing security settings after the CIFS server is stopped](#)
- [About Active Directory \(AD\)](#)
- [About configuring CIFS for Active Directory \(AD\) domain mode](#)
- [Setting NTLM](#)
- [About setting trusted domains](#)
- [About storing account information](#)
- [Storing user and group accounts](#)
- [Reconfiguring the CIFS service](#)
- [About mapping user names for CIFS/NFS sharing](#)
- [About the mapuser commands](#)
- [Adding, removing, or displaying the mapping between CIFS and NFS users](#)
- [Automatically mapping of UNIX users from LDAP to Windows users](#)

- [About managing home directories](#)
- [About CIFS clustering modes](#)
- [About migrating CIFS shares and home directories](#)
- [Setting the CIFS aio_fork option](#)
- [About managing local users and groups](#)
- [Enabling CIFS data migration](#)

About configuring Veritas Access for CIFS

The Common Internet File System (CIFS), also known as the Server Message Block (SMB), is a network file sharing protocol that is widely used on Microsoft and other operating systems. Veritas Access supports the SMB3 protocol.

You can specify either an IPv4 address or an IPv6 address.

Veritas Access supports the following clustering modes:

- Normal
- Clustered Trivial Database (CTDB) - a cluster implementation of the TDB (Trivial database) based on the Berkeley database API

You can configure Active Directory by navigating to **Settings > User Management > Active Directory Management**.

Veritas Access supports the following CIFS security modes:

- User
- ADS

Each clustering mode supports both of the CIFS security modes. The ctdb clustering mode is a different clustered implementation of Veritas Access CIFS, which supports almost all of the features supported by normal clustering mode as well as some additional features.

Additional features supported in ctdb clustering mode:

- Directory-level share support and also supported in normal clustering mode
- Multi-instance share export of a file system/directory
- Simultaneous access of a share from multiple nodes and therefore better load balancing

See [“About CIFS clustering modes”](#) on page 159.

Veritas Access can be integrated into a network that consists of machines running Microsoft Windows. You can control and manage the network resources by using Active Directory (AD) domain controllers.

Before you use Veritas Access with CIFS, you must have administrator-level knowledge of the Microsoft operating systems, Microsoft services, and Microsoft protocols (including AD and NT services and protocols).

You can find more information about them at: www.microsoft.com.

When serving the CIFS clients, Veritas Access can be configured to operate in one of the operating mode environments described in [Table 10-1](#).

Table 10-1 CIFS operating mode environments

Mode	Definition
Standalone	Information about the user and group accounts is stored locally on Veritas Access. Veritas Access also authenticates users locally using the Linux password and group files. This mode of operation is provided for Veritas Access testing and may be appropriate in other cases, for example, when Veritas Access is used in a small network and is not a member of a Windows security domain. In this mode of operation, you must create the local users and groups; they can access the shared resources subject to authorization control.
Active Directory (AD)	Veritas Access becomes a member of an AD security domain and is configured to use the services of the AD domain controller, such as DNS, LDAP, and NTP. Kerberos, NTLMv2, or NTLM authenticate users.

When Veritas Access operates in the AD domain mode, it acts as a domain member server and not as the domain controller.

About configuring CIFS for standalone mode

If you do not have an AD server, you can use Veritas Access as a standalone server. Veritas Access is used in standalone mode when testing Veritas Access functionality and when it is not a member of a domain.

Before you configure the CIFS service for the standalone mode, do the following:

- Make sure that the CIFS server is not running.
- Set security to user.
- Start the CIFS server.

To make sure that the configuration has changed, do the following:

- Check the server status.

- Display the server settings.

Configuring CIFS server status for standalone mode

To check the CIFS server status

- 1 To check the status of the CIFS server, enter the following:

```
CIFS> server status
```

By default, `security` is set to `user`, the required setting for standalone mode. The following example shows that `security` was previously set to `ads`.

- 2 If the server is running, enter the following:

```
CIFS> server stop.
```

To check the security setting

- 1 To check the current settings before setting security, enter the following:

```
CIFS> show
```

- 2 To set security to `user`, enter the following:

```
CIFS> set security user
```

To start the CIFS service in standalone mode

- 1 To start the service in standalone mode, enter the following:

```
CIFS> server start
```

- 2 To display the new settings, enter the following:

```
CIFS> show
```

- 3 To make sure that the server is running in standalone mode, enter the following:

```
CIFS> server status
```

The CIFS service is now running in standalone mode.

See [“About managing local users and groups”](#) on page 163.

See [“About managing CIFS shares”](#) on page 288.

Changing security settings

To change security settings

- ◆ To set the security to user, enter the following:

```
CIFS> set security user
```

To stop the CIFS server:

```
CIFS> server stop
```

Changing security settings after the CIFS server is stopped

To change security settings for a CIFS server that has been stopped

- ◆ To set security to a value other than `domain`, enter the following:

```
CIFS> set security user
```

If the server is stopped, then changing the security mode will disable the membership of the existing domain.

About Active Directory (AD)

In order to provide CIFS services, Veritas Access must be able to authenticate within the Windows environment.

Active Directory (AD) is a technology created by Microsoft that provides a variety of network services including LDAP directory services, Kerberos-based authentication, Domain Name System (DNS) naming, secure access to resources, and more.

Veritas Access will not join the AD domain if its clock is excessively out-of-sync with the clock on the AD domain controller. Ensure that Network Time Protocol (NTP) is configured on Veritas Access, preferably on the same NTP server as the AD domain controller.

Configuring entries for Veritas Access DNS for authenticating to Active Directory (AD)

Name resolution must be configured correctly on Veritas Access. Domain Name System (DNS) is usually used for name resolution.

To configure entries for Veritas Access DNS for authenticating to Active Directory

1 Create an entry for the Veritas Access cluster name.

The cluster name is chosen at the time of installation, and it cannot be reset afterwards. It is also the NetBios name of the cluster, hence it must resolve to an IP address.

2 Configure the Veritas Access cluster name in DNS so that queries to it return the Virtual IP Addresses (VIPs) associated with the Veritas Access cluster in a round-robin fashion.

This is done by creating separate A records that map the cluster name to each VIP. So, if there are four VIPs associated with the Veritas Access cluster (not including special VIPs for backup, replication for Veritas Access, and so on), then there must be four A records mapping the cluster name to the four VIPs.

3 Verify that the DNS server has correct entries for Veritas Access by querying from a client:

```
myclient:~ # nslookup myaccess
```

After configuring the DNS server correctly, Veritas Access must be configured as a DNS client.

This is done during installation, but may be modified by using the following commands:

```
Network> dns set domainname accesstest-ad2.local
```

```
Network> dns set nameservers <IP address>
```

```
Network> dns enable
```

4 Verify that DNS client parameters are set correctly by entering the following command:

```
Network> dns show
```

5 Ensure host resolution is querying DNS by checking nsswitch:

```
Network> nsswitch show
```

In the above scenario, host resolution first looks at files, and then DNS.

Configuring name resolution correctly is critical in order to successfully join Veritas Access to Active Directory.

Joining Veritas Access to Active Directory (AD)

To join Veritas Access to Active Directory (AD)

- 1 To stop the CIFS server, enter the following command.

```
CIFS> server stop
```

- 2 To set the domain, enter the following command:

```
CIFS> set domain accesstest-ad2.local
```

In this example, it is the same as the DNS domain name.

This is the domain name of Active Directory.

- 3 To set the domain controller, enter the following command:

```
CIFS> set domaincontroller <IP address>
```

The IP address can be the IP address of the Active Directory Domain Controller. However, this is not a requirement. The DNS server and Active Directory can run on different servers, and hence this IP address may be different from the IP address of the DNS server.

- 4 To set the domain user, enter the following command:

```
CIFS> set domainuser newuser
```

This is a user whose credentials are used to join the Active Directory domain. The `domainuser` must have Domain Join privilege into the Active Directory domain. The `domainuser` need not be Administrator.

- 5 To set the CIFS security mode, enter the following command:

```
CIFS> set security ads
```

The other CIFS security mode is `user` for local users. For authenticating to Active Directory, use the `ads` CIFS security mode.

- 6 To start the CIFS server, enter the following command:

```
CIFS> server start
```

Veritas Access displays the time on the cluster as well as the time on the Active Directory Domain Controller.

If NTP has been configured correctly, then there will be no time skew.

Otherwise, you will need to reconfigure NTP correctly.

You will be prompted to enter the password of `domainuser`.

Verifying that Veritas Access has joined Active Directory (AD) successfully

To verify that Veritas Access has joined Active Directory (AD) successfully

- ◆ To verify that Veritas Access has joined Active Directory successfully, enter the following command:

```
CIFS> server status
```

Refer to the `Domain membership status` line of the output to verify that the Veritas Access cluster has joined the domain (displays as `Enabled`) if the join is successful.

If the cluster did not join the domain, an informative error message is provided indicating why the Veritas Access cluster cannot join the domain.

About configuring CIFS for Active Directory (AD) domain mode

This section assumes that an Active Directory (AD) domain has already been configured and that Veritas Access can communicate with the AD domain controller (DC) over the network. The AD domain controller is also referred to as the AD server.

Configuring CIFS for the AD domain mode

To set the domain user for AD domain mode

- 1 To verify that the CIFS server is stopped, enter the following:

```
CIFS> server status
```

- 2 If the server is running, stop the server. Enter the following:

```
CIFS> server stop
```

- 3 To set the domain user, enter the following:

```
CIFS> set domainuser username
```

where *username* is the name of an existing AD domain user who has permission to perform the join domain operation.

For example:

```
CIFS> set domainuser administrator
```

To set the domain for AD domain mode

- ◆ To set the domain for AD domain mode, enter the following:

```
CIFS> set domain domainname
```

where *domainname* is the name of the domain.

For example:

```
CIFS> set domain VERITASDOMAIN.COM
```

To set the domain controller for AD domain mode

- ◆ To set the domain controller, enter the following:

```
CIFS> set domaincontroller servername
```

where *servername* is the server's IP address or DNS name.

To set security to ads

- ◆ To set security to ads, enter the following:

```
CIFS> set security ads|user
```

Enter `ads` for *security*.

```
CIFS> set security ads
```

To set the workgroup

- ◆ To set the workgroup name if the WORKGROUP or NetBIOS domain name is different from the domain name, enter the following:

```
CIFS> set workgroup workgroup
```

where *workgroup* sets the WORKGROUP name. If the name of the WORKGROUP or NetBIOS domain name is different from the domain name, use this command to set the WORKGROUP name.

For example, if SIMPLE is the name of the WORKGROUP you want to set, you would enter the following:

```
CIFS> set workgroup SIMPLE
```

Though the following symbols \$, (), ', and & are valid characters for naming a WORKGROUP, the Veritas Access CIFS implementation does not allow using these symbols.

To start the CIFS server

- 1 To start the CIFS server, enter the following:

```
CIFS> server start
```

After you enter the correct password for the user administrator belonging to AD domain, you get a message saying that the CIFS server has started successfully.

- 2 To make sure that the service is running, enter the following:

```
CIFS> server status
```

The CIFS server is now running in the AD domain mode. You can export the shares, and the domain users can access the shares subject to the AD authentication and authorization control.

Using multi-domain controller support in CIFS

Veritas Access allows you to set a comma-separated list of primary and backup domain controllers for the given domain.

Note: You need to set dns nameserver for other domain controller (i.e. backup domain controller) using the `network dns set nameserver` command.

You will need to stop and start the CIFS server.

See [“Reconfiguring the CIFS service”](#) on page 149.

To display the list of domain controllers

- ◆ To display the list of domain controllers, enter the following:

```
CIFS> show
```

If the primary domain controller goes down, the CIFS server tries the next domain controller in the list until it receives a response. You should always point Veritas Access to the trusted domain controllers to avoid any security issues. Veritas Access does not perform list reduction or reordering, instead it uses the list as it is. So, avoid entering the redundant name for the same domain controller.

About leaving an AD domain

There is no Veritas Access command that lets you leave an AD domain. It happens automatically as a part of change in security or domain settings, and then starts or stops the CIFS server. Thus, Veritas Access provides the domain leave operation depending on existing security and domain settings and new administrative commands. However, the leave operation requires the credentials of the old domain's user. All of the cases for a domain leave operation have been documented in [Table 10-2](#).

Table 10-2 Change AD domain mode settings commands

Command	Definition
<code>set domain</code>	<p>Sets the domain.</p> <p>When you change any of the domain settings and you restart the CIFS server, the CIFS server leaves the old domain. Thus, when a change is made to either one or more of domain, domain controller, or domain user settings, and the next time the CIFS server is started, the CIFS server first attempts to leave the existing join and then joins the AD domain with the new settings.</p> <p>See “About leaving an AD domain” on page 128.</p>
<code>set security user</code>	<p>Sets the security user.</p> <p>If you change the security setting from <code>ads</code> to <code>user</code> and you stop or restart the CIFS server, it leaves the AD domain.</p> <p>When you change the security setting, and you stop or restart the CIFS server, the CIFS server leaves the existing AD domain. For example, the CIFS server leaves the existing AD domain if the existing security is <code>ads</code>, and the new security is changed to <code>user</code>, and the CIFS server is either stopped, or started again.</p> <p>See “About leaving an AD domain” on page 128.</p> <p>If the CIFS server is already stopped, changing the security to a value other than <code>ads</code> causes Veritas Access to leave the domain. Both the methods mentioned earlier require either stopping or starting the CIFS server. This method of leaving the domain is provided so that if a CIFS server is already stopped, and may not be restarted in near future, you should have some way of leaving an existing join to AD domain.</p> <p>See “About leaving an AD domain” on page 128.</p>

Changing domain settings for AD domain mode

Each case assumes that the Veritas Access cluster is part of an AD domain.

To verify the cluster is part of an AD domain

- ◆ To verify that the cluster is part of an AD domain, enter the following:

```
CIFS> server status
```

To change domain settings for AD domain mode

- 1 To stop the CIFS server, enter the following:

```
CIFS> server stop
```

- 2 To change the domain, enter the following:

```
CIFS> set domain newdomain.com
```

When you start the CIFS server, it tries to leave the existing domain. This requires the old domainuser to enter its password. After the password is supplied, and the domain leave operation succeeds, the CIFS server joins an AD domain with the new settings.

- 3 To start the CIFS server, enter the following:

```
CIFS> server start
```

To change the security settings for the AD domain mode

- ◆ To set the security to user, enter the following:

```
CIFS> set security user
```

To stop the CIFS server:

```
CIFS> server stop
```

Changing security settings with stopped server on the AD domain mode

- ◆ To set security to a value other than ads, enter the following:

```
CIFS> set security user
```

Removing the AD interface

You can remove the Veritas Access cluster from the AD domain by using the Active Directory interface.

To remove the Veritas Access cluster

- 1 Open the interface **Active Directory Users and Computers**.
- 2 In the domain hierarchy tree, click on **Computers**.
- 3 In the details pane, right-click the computer entry corresponding to Veritas Access (this can be identified by the Veritas Access cluster name) and click **Delete**.

Setting NTLM

When you use Veritas Access in AD domain mode, there is an optional configuration step that can be done. You can disable the use of Microsoft NTLM (NT LAN Manager) protocol for authenticating users.

When the Veritas Access CIFS service is running in the standalone mode (with security set to user) some versions of the Windows clients require NTLM authentication to be enabled. You can do this by setting `CIFS> set ntlm_auth` to `yes`.

When NTLM is disabled and you use Veritas Access in AD domain mode, the available authentication protocols are Kerberos and NTLMv2. The one used depends on the capabilities of both the Veritas Access clients, and domain controller. If no special action is taken, Veritas Access allows the NTLM protocol to be used.

For any specific CIFS connection, all the participants, that is the client machine, Veritas Access and the domain controller select the protocol that they all support and that provides the highest security. In the AD domain mode, Kerberos provides the highest security.

To disable NTLM

- 1 If the server is running, enter the following:

```
CIFS> server stop
```

- 2 To disable NTLM, enter the following:

```
CIFS> set ntlm_auth no
```

- 3 To start the CIFS service, enter the following:

```
CIFS> server start
```

To enable NTLM

- 1 If the server is running, enter the following:

```
CIFS> server stop
```

- 2 To enable the NTLM protocol, enter the following:

```
CIFS> set ntlm_auth yes
```

- 3 To start the CIFS service, enter the following:

```
CIFS> server start
```

About setting trusted domains

The Microsoft Active Directory supports the concept of trusted domains. When you authenticate users, you can configure domain controllers in one domain to trust the domain controllers in another domain. This establishes the trust relation between the two domains. When Veritas Access is a member in an AD domain, both Veritas Access and the domain controller are involved in authenticating the clients. You can configure Veritas Access to support or not support trusted domains.

Table 10-3 Set trusted domains commands

Command	Definition
<code>set allow_trusted_domains yes</code>	Enables the use of trusted domains in the AD domain mode. Note: If the security mode is <code>user</code> , it is not possible to enable AD trusted domains. All the IDMAP backend methods (<code>rid</code> , <code>ldap</code> , and <code>hash</code>) are able to support trusted domains. See “Setting Active Directory trusted domains” on page 144.
<code>set allow_trusted_domains no</code>	Disables the use of trusted domains in the AD domain mode. See “Setting Active Directory trusted domains” on page 144.

Specifying trusted domains that are allowed access to the CIFS server

You can specify the trusted domains that are allowed access to a CIFS server when the `CIFS> set allow_trusted_domains` option is set to `yes`, and `idmap_backend` is set to `rid` or `ad`.

See [“Allowing trusted domains access to CIFS when setting an IDMAP backend to rid”](#) on page 133.

By default, all the trusted domains of the joined active directory domain are included in the CIFS settings and configuration if `allow_trusted_domains` is set to `yes`.

By default, `CIFS> set allow_trusted_domains` is set to `no`.

To specify the trusted domains that are allowed access to the CIFS server

- ◆ To specify the trusted domains that are allowed access to the CIFS server, enter the following:

```
CIFS> set allow_trusted_domains yes | no [trusted_domains]
```

where *trusted_domains* are the trusted domains that you want to allow access to the CIFS server.

Allowing trusted domains access to CIFS when setting an IDMAP backend to rid

To allow trusted domains access to CIFS when setting IDMAP backend to rid

- 1 If the CIFS server is running, enter the following:

```
CIFS> server stop
```

- 2 To set the *idmap_backend* to *rid*, enter the following:

```
CIFS> set idmap_backend rid [uid_range]
```

where *uid_range* represents the range of identifiers that are used by Veritas Access when mapping domain users and groups to local users and groups.

You can obtain unique user IDs (UIDs) or group IDs (GIDs) from domains by reading ID mappings from an Active Directory server that uses RFC2307/SFU schema extensions. This is a read-only idmap backend. Trusted domains are allowed if *set allow_trusted_domains* is set to *yes*. A valid user from a domain or trusted domain should have a UID as well as a GID for the user's primary group.

By default, the *uid_range* is set to 10000-1000000. Change it in cases where there are more than 1,000,000 users existing on a local Veritas Access cluster where there are joined Active Directory domains or trusted domains.

Note: The *uid_range* is adjusted automatically according to the search results of the defined UNIX IDs from the domain after a CIFS server restart.

```
CIFS> set idmap_backend rid
```

- 3 To set *allow_trusted_domains* to *yes*, enter the following:

```
CIFS> set allow_trusted_domains yes
```

- 4 To start the CIFS server again, enter the following:

```
CIFS> server start
```

- 5 To verify the CIFS server status when there are trusted domains, enter the following:

```
CIFS> server status
```

Domain names containing square brackets indicate that the domain used to be a trusted domain, but the domain is currently obsolete.

Allowing trusted domains access to CIFS when setting an IDMAP backend to ldap

To allow trusted domains access to CIFS when setting an IDMAP backend to ldap

- 1 To configure AD as an IDMAP backend, follow the steps provided at:
See [“About configuring Windows Active Directory as an IDMAP backend for CIFS”](#) on page 138.
- 2 To set `idmap_backend` to `ldap`, enter the following:

```
CIFS> set idmap_backend ldap [idmap_ou] [uid_range]
```

`idmap_ou` Specifies the CIFS idmap Organizational Unit Name (OU) configured on the LDAP server, which is used by Veritas Access when mapping users and groups to local users and groups. The default value is `cifsidmap`.

`uid_range` Specifies the range of identifiers that are used by Veritas Access when mapping domain users and groups to local users and groups.

You can obtain unique user IDs (UIDs) or group IDs (GIDs) from domains by reading ID mappings from an Active Directory server that uses RFC2307/SFU schema extensions. This is a read-only idmap backend. Trusted domains are allowed if `set allow_trusted_domains` is set to `yes`. A valid user from a domain or trusted domain should have a UID as well as a GID for the user's primary group.

By default, the `uid_range` is set to 10000-1000000. Change it in cases where there are more than 1,000,000 users existing on a local Veritas Access cluster where there are joined Active Directory domains or trusted domains.

Note: The `uid_range` is adjusted automatically according to the search results of the defined UNIX IDs from the domain after a CIFS server restart.

```
CIFS> set idmap_backend ldap
```

- 3 To set `allow_trusted_domains` to `yes`, enter the following:

```
CIFS> set allow_trusted_domains yes
```

- 4 To restart the CIFS server again, enter the following:

```
CIFS> server start
```

- 5 To verify the CIFS server status when there are trusted domains, enter the following:

```
CIFS> server status
```

Allowing trusted domains access to CIFS when setting an IDMAP backend to hash

To allow trusted domains access to CIFS when setting an IDMAP backend to hash

- 1 If the CIFS server is running, enter the following:

```
CIFS> server stop
```

- 2 To set `idmap_backend` to `hash`, enter the following:

```
CIFS> set idmap_backend hash
```

You can obtain unique user IDs (UIDs) or group IDs (GIDs) from domains by reading ID mappings from an Active Directory server that uses RFC2307/SFU schema extensions. This is a read-only idmap backend. Trusted domains are allowed if `set allow_trusted_domains` is set to `yes`. A valid user from a domain or trusted domain should have a UID as well as a GID for the user's primary group.

By default, the `uid_range` is set to 10000-1000000. Change it in cases where there are more than 1,000,000 users existing on a local Veritas Access cluster where there are joined Active Directory domains or trusted domains.

Note: The `uid_range` is adjusted automatically according to the search results of the defined UNIX IDs from the domain after a CIFS server restart.

- 3 To set `allow_trusted_domains` to `yes`, enter the following:

```
CIFS> set allow_trusted_domains yes
```

- 4 To verify the CIFS server status when there are trusted domains, enter the following:

```
CIFS> server status
```

Allowing trusted domains access to CIFS when setting an IDMAP backend to `ad`

To allow trusted domains access to CIFS when setting IDMAP backend to `ad`

- 1 If the CIFS server is running, enter the following:

```
CIFS> server stop.
```

- 2 To set the `idmap_backend` to `ad`, enter the following:

```
CIFS> set idmap_backend ad [uid_range]
```

where `uid_range` represents the range of identifiers that are used by Veritas Access when mapping domain users and groups to local users and groups.

You can obtain unique user IDs (UIDs) or group IDs (GIDs) from domains by reading ID mappings from an Active Directory server that uses RFC2307/SFU schema extensions. This is a read-only idmap backend. Trusted domains are allowed if `set allow_trusted_domains` is set to `yes`. A valid user from a domain or trusted domain should have a UID as well as a GID for the user's primary group.

By default, the `uid_range` is set to 10000-1000000. Change it in cases where there are more than 1,000,000 users existing on a local `&product_name_isa` cluster where there are joined Active Directory domains or trusted domains.

Note: The `uid_range` is adjusted automatically according to the search results of the defined UNIX IDs from the domain after a CIFS server restart.

- 3 To set `allow_trusted_domains` to `yes`, enter the following:

```
CIFS> set allow_trusted_domains yes
```

- 4 To start the CIFS server again, enter the following:

```
CIFS> server start
```

- 5 To verify the CIFS server status when there are trusted domains, enter the following:

```
CIFS> server status
```

Domain names containing square brackets indicate that the domain used to be a trusted domain, but the domain is currently obsolete.

About configuring Windows Active Directory as an IDMAP backend for CIFS

The CIFS server requires equivalent UNIX identities for Windows accounts to service requests from Windows clients. In the case of trusted domains, Veritas Access has to store the mapped UNIX identities (IDMAP) in a centralized database that is accessible from each of the cluster nodes.

Active Directory (AD), as with any LDAP V3 compliant directory service, can function as the backend for CIFS IDMAP backend storage. When the CIFS server joins a Windows Active Directory Domain as a member server, and you want to use LDAP as an IDMAP backend, then it is necessary to create an Active Directory application partition for the IDMAP database. To support the creation of an Active Directory application partition, Windows 2003 R2 and above version is required.

Active Directory application partition provides the ability to control the scope of replication and allow the placement of replicas in a manner more suitable for dynamic data. As a result, the application directory partition provides the capability of hosting dynamic data in the Active Directory server, thus allowing ADSI/LDAP access to it.

By extending the AD schema with the necessary CIFS-schema extensions, and creating an AD application partition, it is possible to store CIFS IDMAP data entries in AD, using one or more domain controllers as IDMAP LDAP backend servers. Also, it is possible to replicate this information in a simple and controlled manner to a subset of AD domain controllers located either in the same domain or in different domains in the AD forest.

Note: A single domain user account is used, for example, **cifsuser** for setting application partition Access Control List (ACL) settings. Make sure the selected user naming context has no space key inside (for example, **CN=cifsuser1,CN=Users,DC=example,DC=com**). A sample AD server is used, for example, **adserver.example.com**. Use relevant values when configuring your AD server.

Configuring the Active Directory schema with CIFS-schema extensions

To extend the Active Directory schema with the necessary CIFS-schema extensions

- 1 Login with **Schema Admins** privileges on the Active Directory Forest Schema Master domain controller.
- 2 Download `ADCIFSSchema.zip` from the Veritas Access server (`/opt/SYMCsnas/install/ADCIFSSchema.zip`) with software such as `WinSCP.exe`.
- 3 Unzip the file and open each `.ldf` file to perform a search and replace of the string `dc=example,dc=com`, replacing the string with the top-level domain component (that is, `dc=yourdomain,dc=com`) values for the AD forest.
- 4 Install the schema extensions by executing the `schemaupdate.bat` file from the command prompt.

To validate the schema extensions

- 1 Execute `regsvr32 schmmgmt.dll` in a command prompt window to install the Active Directory Schema Snap-In on the AD server.
- 2 Enter `mmc` in **Run**.
- 3 On the **File** menu, click **Add/Remove Snapin**.
- 4 In **Available snap-ins**, click **Active Directory Schema**, and then click **Add**.
- 5 Click **OK**.
- 6 Click **Attributes** in the left frame, and try to find **uidNumber** and **gidNumber** in the right frame.

Validate that the **uidNumber** and **gidNumber** attributes have no minimum or maximum value setting by viewing the properties of the attribute objects.

To create an application partition

- 1 Open a command prompt window on the domain controller that will hold the first replica of the application partition.
- 2 Enter `ntdsutil` in the command prompt window.
- 3 At the `ntdsutil` command prompt, enter the following:

```
domain management
```

If you are using Windows 2008, change this command to the following:

```
partition management
```

- 4** At the domain management command prompt, enter the following:

```
connection
```

- 5** At the connection command prompt, enter the following:

```
connect to server adserver.example.com
```

- 6** At the connection command prompt, enter the following:

```
quit
```

- 7** At the domain management command prompt, enter the following such as:

```
create nc dc=idmap,dc=example,dc=com null
```

Example settings:

```
C:\>ntdsutil
ntdsutil: domain management
domain management: connection
server connections: connect to server adserver.example.com
Binding to adserver.example.com ...
Connected to adserver.si2m.com using credentials of locally logged
on user.
server connections: quit
domain management: create nc dc=idmap,dc=example,dc=com NULL
adding object dc=idmap,dc=example,dc=com
domain management: quit
ntdsutil: quit
Disconnecting from adserver.example.com...
```

- 8 Once the application partition has been created, open **ADSIedit.msc** from **Run**, then right-click on **ADSI Edit** in the left frame, and click **connect to ...** to connect to the application partition using the settings as indicated:

Name	Enter Domain .
Connection Point	Select or enter a Distinguished Name or Naming Context , as in: dc=idmap,dc=example,dc=com
Computer	Select or enter a domain or server, as in: adserver.example.com

Connection Settings

Name:

Path:

Connection Point:

☒ Select or type a Distinguished Name or Naming Context:

☐ Select a well known Naming Context:

Computer:

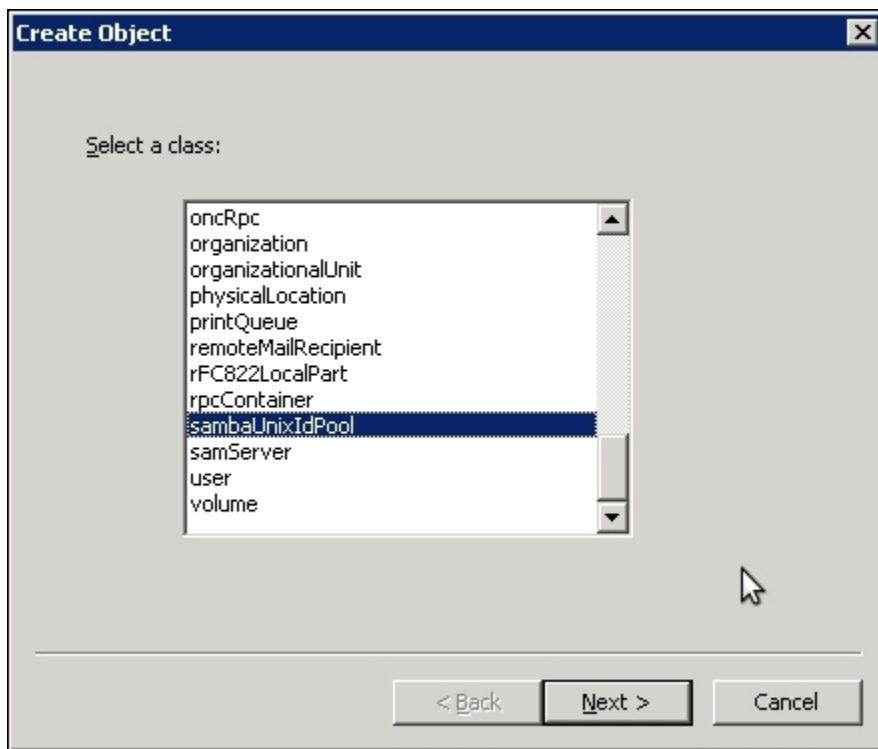
☒ Select or type a domain or server: (Server | Domain [:port])

☐ Default (Domain or server that you logged in to)

☐ Use SSL-based Encryption

Advanced... OK Cancel

- 9 Once connected, select the top-level application partition (for example, **dc=idmap,dc=example,dc=com**) node in the left panel, and right-click to select **New** then **Object** from the list, and then select **SambaUnixIdPool**.



When prompted, enter the following values:

OU attribute	cifsidmap
uidNumber	10000
gidNumber	10000

- 10 Click **Finish** to complete the configuration.
- 11 Once the **ou=cifsidmap,dc=idmap,dc=example,dc=com** container has been created, right-click the object, and select **properties**.
- 12 On the **Security** tab, click **Add**, and proceed to add the cifsuser user account, and grant the account Read, Write, Create All Child Objects, and Delete All Child Objects permissions.

Configuring the LDAP client for authentication using the CLI

To configure the LDAP client for authentication using the CLI

- 1 Log into the cluster CLI using the `master` account.
- 2 Configure `Network> ldap` settings.

Example settings:

```
Network> ldap set basedn dc=idmap,dc=example,dc=com
Network> ldap set binddn cn=cifsuser,dc=example,dc=com
Network> ldap set rootbinddn cn=cifsuser,cn=users,dc=example,dc=com
Network> ldap set server adserver.example.com
Network> ldap enable
```

Configuring the CIFS server with the LDAP backend

To configure the CIFS server with the LDAP backend

- 1 Log in to the Veritas Access cluster CLI using the `master` account.
- 2 Set the domain, domaincontroller, and domainuser.
- 3 Set security to `ads`.

- 4 Set `idmap_backend` to `ldap`, and specify `idmap OU` as `cifsidmap`.

Example settings:

```
CIFS> set domain example.com
CIFS> set domainuser administrator
CIFS> set domaincontroller adserver.example.com
CIFS> set security ads
CIFS> set idmap_backend ldap cifsidmap
CIFS> server start
```

- 5 Start the CIFS server.

The CIFS server will take some time to import all the users from the joined domain and trusted domain(s) to the application partition. Wait for at least ten minutes before trying to access the shares from Windows clients after starting the CIFS server.

To validate that IDMAP entries are being entered correctly in the Active Directory application partition, connect to the Active Directory application partition using an LDAP administration tool, for example, LDP or ADSIEdit. Expand the IDMAP container (`ou=cifsidmap`). There should be numerous entries.

Setting Active Directory trusted domains

To enable Active Directory (AD) trusted domains

- 1 If the server is running, enter the following:

```
CIFS> server stop
```

- 2 To enable trusted domains, enter the following:

```
CIFS> set allow_trusted_domains yes
```

- 3 To start the CIFS server, enter the following:

```
CIFS> server start
```


To disable trusted domains

- 1 If the server is running, enter the following:

```
CIFS> server stop
```

- 2 To disable trusted domains, enter the following:

```
CIFS> set allow_trusted_domains no
```

- 3 To start the CIFS server, enter the following:

```
CIFS> server start
```

About storing account information

Veritas Access maps between the domain users and groups (their identifiers) and local representation of these users and groups. Information about these mappings can be stored locally on Veritas Access or remotely using the DC directory service. Veritas Access uses the `idmap_backend` configuration option to decide where this information is stored.

This option can be set to one of the following:

<code>rid</code>	Maps SIDs for domain users and groups by deriving UID and GID from RID on the Veritas Access CIFS server.
<code>ldap</code>	Stores the user and group information in the LDAP directory service.
<code>hash</code>	Maps SIDs for domain users and groups to 31-bit UID and GID by the implemented hashing algorithm on the Veritas Access CIFS server.
<code>ad</code>	Obtains unique user IDs (UIDs) or group IDs (GIDs) from domains by reading ID mappings from an Active Directory server that uses RFC2307/SFU schema extensions.

Note: SID/RID are Microsoft Windows concepts that can be found at:
[http://msdn.microsoft.com/en-us/library/aa379602\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa379602(VS.85).aspx).

The `rid` and `hash` values can be used in any of the following modes of operation:

- Standalone
- AD domain

`rid` is the default value for `idmap_backend` in all of these operational modes. The `ldap` value can be used if the AD domain mode is used.

When security is set as "user" `idmap_backend` is irrelevant.

Table 10-4 Store account information commands

Command	Definition
<code>set idmap_backend rid</code>	<p>Configures Veritas Access to store information about users and groups locally.</p> <p>Trusted domains are allowed if <code>allow_trusted_domains</code> is set to <code>yes</code>. The <code>uid_range</code> is set to 10000-1000000 by default.</p> <p>Change the default range in cases where it is not appropriate to accommodate local Veritas Access cluster users, Active Directory, or trusted domain users.</p> <p>Do not attempt to modify <code>LOW_RANGE_ID</code> (10000) if user data has already been created or copied on the CIFS server. This may lead to data access denied issues since the UID changes.</p> <p>See "Storing user and group accounts" on page 147.</p>
<code>set idmap_backend hash</code>	<p>Allows you to obtain the unique SID to UID/GID mappings by the implemented hashing algorithm. Trusted domains are allowed if <code>allow_trusted_domains</code> is set to <code>yes</code>.</p> <p>See "Storing user and group accounts" on page 147.</p>
<code>set idmap_backend ad</code>	<p>Allows you to obtain unique user IDs (UIDs) or group IDs (GIDs) from domains by reading ID mappings from an Active Directory server that uses RFC2307/SFU schema extensions.</p> <p>See "Storing user and group accounts" on page 147.</p>

Table 10-4 Store account information commands (*continued*)

Command	Definition
<code>set</code> <code>idmap_backend</code> <code>ldap</code>	<p>Configures Veritas Access to store information about users and groups in a remote LDAP service. You can only use this command when Veritas Access is operating in the AD domain mode. The LDAP service can run on the domain controller or it can be external to the domain controller.</p> <p>Note: For Veritas Access to use the LDAP service, the LDAP service must include both RFC 2307 and proper schema extensions.</p> <p>See “Configuring the LDAP client for authentication using the CLI” on page 143.</p> <p>This option tells the CIFS server to obtain SID to UID/GID mappings from a common LDAP backend. This option is compatible with multiple domain environments. So <code>allow_trusted_domains</code> can be set to <code>yes</code>.</p> <p>If <code>idmap_backend</code> is set to <code>ldap</code>, you must first configure the Veritas Access LDAP options using the <code>Network> ldap</code> commands.</p> <p>See “About configuring LDAP settings” on page 57.</p> <p>See “Storing user and group accounts” on page 147.</p>

Storing user and group accounts

To set `idmap_backend` to `rid`

- 1 If the server is running, enter the following:

```
CIFS> server stop
```

- 2 To store information about user and group accounts locally, enter the following:

```
CIFS> set idmap_backend rid [uid_range]
```

where `uid_range` represents the range of identifiers that are used by Veritas Access when mapping domain users and groups to local users and groups. The default range is `10000-1000000`.

- 3 To start the CIFS server, enter the following:

```
CIFS> server start
```

To set `idmap_backend` to LDAP

- 1 To make sure that you have first configured LDAP, enter the following:

```
Network> ldap show
```

- 2 If the CIFS server is running, enter the following:

```
CIFS> server stop
```

- 3 To use the remote LDAP store for information about the user and group accounts, enter the following:

```
CIFS> set idmap_backend ldap [idmap_ou]
```

where *idmap_ou* represents the CIFS idmap Organizational Unit Name (OU) configured on the LDAP server, which is used by Veritas Access when mapping users and groups to local users and groups. The default value is `cifsidmap`.

- 4 To start the CIFS server, enter the following:

```
CIFS> server start
```

To set `idmap_backend` to a hash algorithm

- 1 If the CIFS server is running, enter the following:

```
CIFS> server stop
```

- 2 To store information about user and group accounts locally, enter the following:

```
CIFS> set idmap_backend hash
```

- 3 To start the CIFS server, enter the following:

```
CIFS> server start
```

To set `idmap_backend` to `ad`

- 1 If the CIFS server is running, enter the following:

```
CIFS> server stop
```

- 2 To obtain the unique UID/GID from domains by reading ID mappings from an Active Directory (AD) server, enter the following:

```
CIFS> set idmap_backend ad [uid_range]
```

where `uid_range` represents the range of identifiers that are used by Veritas Access when mapping domain users and groups to local users and groups. The default range is `10000-1000000`. Change it in cases where there are more than 1,000,000 users existing on a local Veritas Access cluster where there are joined Active Directory domains or trusted domains.

Note: The `uid_range` is adjusted automatically according to the search results of the defined UNIX IDs from the domain after a CIFS server restart.

- 3 To start the CIFS server, enter the following:

```
CIFS> server start
```

Reconfiguring the CIFS service

Sometime after you have configured the CIFS service, and used it for awhile, you need to change some of the settings. For example, you may want to allow the use of trusted domains or you need to move Veritas Access from one security domain to another. To carry out these changes, set the new settings and then start the CIFS server. As a general rule, you should stop the CIFS service before making the changes.

An example where Veritas Access is moved to a new security domain (while the mode of operation stays unchanged as, AD domain) is referenced below.

This example deals with reconfiguring CIFS. So make sure that if any of the other AD services like DNS or NTP are being used by Veritas Access, that Veritas Access has already been configured to use these services from the AD server belonging to the new domain.

Make sure that the DNS service, NTP service and, if used as an ID mapping store, also the LDAP service, are configured as required for the new domain.

To reconfigure the CIFS service, do the following:

- Make sure that the server is not running.
- Set the domain user, domain, and domain controller.
- Start the CIFS server.

To set the user name for the AD

- 1 To verify that the CIFS server is stopped, enter the following:

```
CIFS> server status
```

- 2 If the server is running, stop the server, and enter the following:

```
CIFS> server stop
```

- 3 To set the user name for the AD, enter the following:

```
CIFS> set domainuser username
```

where *username* is the name of an existing AD domain user who has permission to perform the join domain operation.

To set the AD domain

- ◆ To set the AD domain, enter the following:

```
CIFS> set domain domainname
```

where *domainname* is the name of the domain. This command also sets the system workgroup. For example:

To set the AD server

- ◆ To set the AD server, enter the following:

```
CIFS> set domaincontroller servername
```

where *servername* is the AD server IP address or DNS name.

If you use the AD server name, you must configure Veritas Access to use a DNS server that can resolve this name.

To start the CIFS server

- 1 To start the CIFS server, enter the following:

```
CIFS> server start
```

- 2 To make sure that the service is running, enter the following:

```
CIFS> server status
```

- 3 To find the current settings, enter the following:

```
CIFS> show
```

About mapping user names for CIFS/NFS sharing

The CIFS server uses user name mapping to translate login names sent by a Windows client to local or remote UNIX user names. The CIFS server uses file lookup for mapping, and this mapping is unidirectional. You can map a CIFS user to an NFS user, but the reverse operation is not possible.

This functionality can be used for the following purposes:

- CIFS and NFS sharing by mapping CIFS users to NFS users
- File sharing among CIFS users by mapping multiple CIFS users to a single UNIX user
- Mapping between two UNIX users by using the `CIFS> mapuser add <CIFSusername> LOCAL <NFSusername>` command, where both the CIFS user and the NFS user are UNIX users

User name mapping is stored in a configuration file.

When user name mapping takes place is dependent on the current security configurations. If security is set to `user`, mapping is done prior to authentication, and a password must be provided for the mapped user name. For example, if there is a mapping between the users `CIFSuser1` and `NFSuser1`. If `CIFSuser1` wants to connect to the Veritas Access server, then `CIFSuser1` needs to provide a password for `NFSuser1`. In this case, `NFSuser1` must be the CIFS local user.

If security is set to either `ads` or `domain`, user name mapping is done after authentication with the domain controller. This means, the actual password must be supplied for the login user `CIFSuser1` in the example cited above. In this case, `NFSuser1` may not be the CIFS local user.

The domain you specify for CIFS user name mapping must be the netbios domain name (instead of the Active Directory DNS domain name) for the user. For example, a netbios domain name might be listed as `VERITASDOMAIN` instead of `VERITASDOMAIN.COM` (without the `.com` extension).

To determine the netbios domain name, login to your Active Directory Server and type the following in a command window:

```
set | findstr DOMAIN
```

The results will include:

```
USERDOMAIN netbios_domain_name  
USERDNSDOMAIN Active_Directory_DNS_domain_name
```

Use the value of `USERDOMAIN` (the netbios domain name) when you map user names.

Note: When setting quotas on home directories and using user name mapping, make sure to set the quota on the home directory using the user name to which the original name is mapped.

Note: For mapped Active Directory users to access their home directory CIFS shares, use the following convention: `\\access\realADuser` instead of `\\access\homes`.

Note: For UNIX users (LDAP/NIS/local) users, make sure to set up these users properly, so that these users are recognized by Samba. User mapping can work properly only after these users are recognized by Samba.

About the mapuser commands

The `CIFS> mapuser` commands are used to add, remove, or display the mapping between CIFS and NFS users.

Typically, a *CIFSusername* is a user coming from an AD server (with a specified domainname), or a locally created CIFS user on this system (local). An *NFSusername* is a user coming from a locally-created CIFS user on this system, or from a NIS/LDAP server configured in the network section.

Note: To make sure user mappings work correctly with a NIS/LDAP server, `Network> nsswitch` settings may need to be adjusted in the `Network> nsswitch` section. You may need to move the position of `ldap` or `nis` in the `Network> nsswitch` section, depending on which name service is being used first.

Adding, removing, or displaying the mapping between CIFS and NFS users

To add a mapping between a CIFS and an NFS user

- ◆ To add a mapping between a CIFS and an NFS user, enter the following:

```
CIFS> mapuser add CIFSusername domainname NFSusername
```

To remove a mapping between a CIFS and an NFS user

- ◆ To remove a mapping between a CIFS and an NFS user, enter the following:

```
CIFS> mapuser remove CIFSusername [domainname]
```

To display a mapping between a CIFS and an NFS user

- ◆ To display a mapping between a CIFS and an NFS user, enter the following:

```
CIFS> mapuser show [CIFSusername] [domainname]
```

Automatically mapping of UNIX users from LDAP to Windows users

To automatically map UNIX users from LDAP to Windows users

- 1 Ensure that Veritas Access joins the LDAP domain using **network ldap**.

From the LDAP server, users and groups should be visible by using the `getent passwd` or `getent group` commands .

- 2 Ensure that Veritas Access joins the Windows AD domain using **cifs**.

- 3 Use the wildcard mapping rule `CIFS> mapuser add * AD Domain Name *`.

The effect is whenever a Windows domain user, say `DOM\foobar`, wants to access CIFS shares, the CIFS server determines if there is a local (a non-Windows) user also named `foobar`, and establishes the mapping between the Windows user and the non-Windows user.

The user name must match between the LDAP and AD domains.

About managing home directories

You can use Veritas Access to store the home directories of CIFS users.

The home directory share name is identical to the Veritas Access user name. When Veritas Access receives a new CIFS connection request, it checks if the requested share is one of the ordinary exported shares. If it is not, Veritas Access checks if the requested share name is the name of an existing Veritas Access user (either local user or domain user, depending on the current mode of operation). If a match is found, it means that the received connection request is for a home directory share.

You can access your home directory share the same way you access the file system ordinary shares. A user can connect only to his or her own home directory.

Note: The internal directories structure of home directory file systems is maintained by Veritas Access. It is recommended not to use a file system as a `homedirfs` that has been used by a normal share in the past or vice versa.

Note: The `CIFS> homedir` commands are supported only from CLISH.

Setting the home directory file systems

Home directory shares are stored in one or more file systems. A single home directory can exist only in one of these file systems, but a number of home directories can exist in a single home directory file system. File systems that are to be used for home directories are specified using the `CIFS> set homedirfs` command.

When a file system is exported as a homedirfs, its mode is set to a 0755 value. This takes place when you start the CIFS server after setting the homedirfs list.

Note: Snapshots cannot be shared as home directory file systems.

To specify one or more file systems as the home directories

- 1 To reserve one or more file systems for home directories, enter the following:

```
CIFS> set homedirfs [filesystemlist]
```

where *filesystemlist* is a comma-separated list of names of the file systems which are used for the home directories.

- 2 If you want to remove the file systems you previously set up, enter the command again, without any file systems:

```
CIFS> set homedirfs
```

- 3 To find which file systems (if any) are currently used for home directories, enter the following:

```
CIFS> show
```

After you select one or more of the file systems to be used in this way, you cannot export the same file systems as ordinary CIFS shares.

If you want to change the current selection, for example, to add an additional file system to the list of home directory file systems or to specify that no file system should be used for home directories, you have to use the same `CIFS> set homedirfs` command. In each case you must enter the entire new list of home directory file systems, which may be an empty list when no home directory file systems are required.

Veritas Access treats home directories differently from ordinary shares. The differences are as follows:

- An ordinary share is used to export a file system, while a number of home directories can be stored in a single file system.

- The file systems used for home directories cannot be exported as ordinary shares.
- Exporting a home directory share is done differently than exporting an ordinary share. Also, removing these two kinds of shares is done differently.
- The configuration options you specify for an ordinary share (such as read-only or use of opportunistic locks) are different from the ones you specify for a home directory share.

Setting up home directories

You can set the home directory for the specified user with the `CIFS> homedir set` command. If the home directory does not exist for the specified user, the `CIFS> homedir set` command creates that user's home directory.

Use the `Storage> quota cifshomedir set` command to set the quota value for the specified user. Otherwise, the value set from the `Storage> quota cifshomedir setdefault` command is used to configure the quota limit. If either the user or default quota is not set, 0 is used as the default value for the unlimited quota.

Once the global quota value is specified, the value applies to the automatically created homedir. For example, if you set the global quota value to `Storage> quota cifshomedir setdefault 100M`, and you then create a new homedir in Windows, then the 100M quota value is assigned to that homedir.

To set the home directory for the specified user

- 1 To set the home directory for the specified user, enter the following:

```
CIFS> homedir set username [domainname] [fsname]
```

<i>username</i>	The name of the CIFS user. If a CIFS user name includes a space, enter the user name with double quotes.
<i>domainname</i>	The domain for the new home directory.
<i>fsname</i>	The home directory file system where the user's home directory is created. If no file system is specified, the user's home directory is created on the home directory file system that has the fewest home directories.

- 2 To find the current settings for a home directory, enter the following:

```
CIFS> homedir show [username] [domainname]
```

<i>username</i>	The name of the CIFS user. If a CIFS user name includes a space, enter the user name with double quotes.
<i>domainname</i>	The Active Directory/Windows NT domain name or specify <code>local</code> for the Veritas Access local user <code>local</code> .

- 3 To find the current settings for all home directories, enter the following:

```
CIFS> homedir show
```

Because the `CIFS> homedir show` command takes a long time when there are more than 1000 CIFS home directories to display, you will be prompted if you want to continue displaying CIFS home directories or not.

When you connect to your home directory for the first time, and if the home directory has not already been created, Veritas Access selects one of the available home directory file systems and creates the home directory there. The file system is selected in a way that tries to keep the number of home directories balanced across all available home directory file systems. The automatic creation of a home directory does not require any commands, and is transparent to both the users and the Veritas Access administrators.

The quota limits the amount of disk space you can allocate for the files in a home directory.

You can set the same quota value for all home directories using the `Storage> quota cifshomedir setall` command.

Displaying home directory usage information

You can display information about home directories using the `CIFS> homedir show` command.

Note: Information about home directory quotas is up-to-date only when you enable the use of quotas for the home directory file systems.

To display information about home directories

- 1 To display information about a specific user's home directory, enter the following:

```
CIFS> homedir show [username] [domainname]
```

username The name of the CIFS user. If a CIFS user name includes a space, enter the user name with double quotes.

domainname The domain where the home directory is located.

- 2 To display information about all home directories, enter the following:

```
CIFS> homedir show
```

Deleting home directories and disabling creation of home directories

You can delete a home directory share. This also deletes the files and sub-directories in the share.

After a home directory is deleted, if you try to access the same home directory again, a new home directory will automatically be created.

If you have an open file when the home directory is deleted, and you try to save the file, a warning appears:

```
Warning: Make sure the path or filename is correct.
```

```
Save dialog?
```

Click on the `Save` button which saves the file to a new home directory.

To delete a home directory share

- ◆ To delete the home directory of a specific user, enter the following:

```
CIFS> homedir delete username [domainname]
```

username The name of the CIFS user. If a CIFS user name includes a space, enter the user name with double quotes.

Respond with *y(es)* or *n(o)* to confirm the deletion.

domainname The domain it is located in.

You can delete all of the home directory shares with the `CIFS> homedir deleteall` command. This also deletes all files and subdirectories in these shares.

After you delete the existing home directories, you can again create the home directories manually or automatically.

To delete the home directories

- ◆ To delete all home directories, enter the following:

```
CIFS> homedir deleteall
```

Respond with *y(es)* or *n(o)* to confirm the deletion.

After you delete the home directories, you can stop Veritas Access serving home directories by using the `CIFS> set homedirfs` command.

To disable creation of home directories

- ◆ To specify that there are no home directory file systems, enter the following:

```
CIFS> set homedirfs
```

After these steps, Veritas Access does not serve home directories.

About CIFS clustering modes

The following clustering modes are supported by Veritas Access:

- Normal
- Clustered Trivial Database (CTDB) - a cluster implementation of the TDB (Trivial database) based on the Berkeley database API

The following operating modes are supported by Veritas Access:

- User

- Domain
- ADS

Each clustering mode supports all of the three operating modes. The ctdb clustering mode is a different clustered implementation of Veritas Access CIFS, which supports almost all of the features that are supported by normal clustering mode as well as some additional features.

Additional features supported in ctdb clustering mode:

- Directory-level share support
- Multi-instance share export of a file system/directory
- Simultaneous access of a share from multiple nodes and therefore better load balancing

About switching the clustering mode

You can switch from normal to ctdb clustering mode or from ctdb to normal clustering mode. You must stop the CIFS server prior to switching to any cluster mode.

About migrating CIFS shares and home directories

You can migrate CIFS shares and home directories from normal to ctdb clustering mode and from ctdb to normal clustering mode.

Veritas Access automatically migrates all CIFS shares and home directories while switching from one clustering mode to another. However, it is not possible to migrate directory-level shares in the normal clustering mode, because directory-level sharing is not supported in normal clustering mode.

Automatic migration of the content of users (that is, users' home directories) from one file system to another file system while switching home directories is not supported. So, if a Veritas Access administrator changes home directories from fs1 to fs2, then users' home directories are not migrated from fs1 to fs2 automatically.

While migrating from normal to ctdb clustering mode, a simple share is created for each split share, because splitting shares is not supported in ctdb clustering mode.

Migrating CIFS shares and home directories from normal to ctdb clustering mode

To migrate CIFS shares and home directories from normal to ctdb clustering mode

- 1 To check the CIFS server status to confirm that the current cluster mode is set to normal, enter the following:

```
CIFS> server status
```

- 2 To list the CIFS shares and home directories, enter the following:

```
CIFS> share show
```

- 3 To stop the CIFS server before changing the clustering mode to ctdb, enter the following:

```
CIFS> server stop
```

```
CIFS> set clustering_mode ctdb
```

- 4 To start the CIFS server in ctdb clustering mode and check the CIFS server status, enter the following:

```
CIFS> server start
```

```
CIFS> server status
```

- 5 To verify that all the CIFS shares and home directories are properly migrated to the ctdb clustering mode, enter the following:

```
CIFS> share show
```

```
CIFS> homedir show
```

Migrating CIFS shares and home directories from ctdb to normal clustering mode

If a file system is exported as multiple CIFS shares in ctdb clustering mode, then while migrating to normal clustering mode, Veritas Access creates only one CIFS share, whichever comes first in the list.

To migrate a CIFS share and home directory from ctldb to normal clustering mode

- 1 To check the status of the CIFS server, enter the following:

```
CIFS> server status
```

- 2 To list the CIFS shares and home directories, enter the following:

```
CIFS> share show
```

```
CIFS> homedir show
```

- 3 To stop the CIFS server to switch the clustering mode to normal, enter the following:

```
CIFS> server stop
```

```
CIFS> set clustering_mode normal
```

- 4 To start the CIFS server in normal clustering mode, enter the following:

```
CIFS> server start
```

If you get a warning message, it indicates that Veritas Access was unable to migrate the directory-level share to normal clustering mode. The rest of the CIFS share and home directory were migrated.

- 5 To list the CIFS shares and home directories after migrating to normal clustering mode, enter the following:

```
CIFS> share show
```

```
CIFS> homedir show
```

Setting the CIFS aio_fork option

The `CIFS> set aio_size` option allows you to set an Asynchronous I/O (AIO) read/write size with an unsigned integer.

To set the `aio_fork` option

- ◆ To set the `aio_fork` option, enter the following:

```
CIFS> set aio_size size
```

where *size* is the AIO read/write size.

If *size* is not set to 0, then enable the `aio_fork` option, and set it as an AIO read/write size. If *size* is set to 0, then disable the `aio_fork` option, and set 0 to an AIO read/write size.

About managing local users and groups

When Veritas Access is operating in the standalone mode, only the local users and groups of users can establish CIFS connections and access the home directories and ordinary shares. The Veritas Access local files store the information about these user and group accounts. Local procedures authenticate and authorize these users and groups based on the use of names and passwords. You can manage the local users and groups as described in the rest of this topic.

Accounts for local users can be created, deleted, and information about them can be displayed using the `CIFS> local user` commands.

Creating a local CIFS user

To create the new local CIFS user

- ◆ To create a local CIFS user, enter the following:

```
CIFS> local user add username [groupelist]
```

where *username* is the name of the user. The *groupelist* is a comma-separated list of group names.

To set the local user password

- ◆ To set the local password, enter the following:

```
CIFS> local password username
```

where *username* is the name of the user whose password you are changing.

To display the local CIFS user(s)

- 1 To display local CIFS users, enter the following:

```
CIFS> local user show [username]
```

where *username* is the name of the user.

- 2 To display one local user, enter the following:

```
CIFS> local user show usr1
```

To delete the local CIFS user

- ◆ To delete a local CIFS user, enter the following:

```
CIFS> local user delete username
```

where *username* is the name of the local user you want to delete.

To change a user's group membership

- ◆ To change a user's group membership, enter the following:

```
CIFS> local user members username grouplist
```

where *username* is the local user name being added to the *grouplist*. Group names in the *grouplist* must be separated by commas.

Configuring a local group

A local user can be a member of one or more local groups. This group membership is used in the standalone mode to determine if the given user can perform some file operations on an exported share. You can create, delete, and display information about local groups using the `CIFS> local group` command.

To create a local group

- ◆ To create a local group, enter the following:

```
CIFS> local group add groupname
```

where *groupname* is the name of the local group.

To list all local groups

- ◆ To list all existing local groups, enter the following:

```
CIFS> local group show [groupname]
```

where *groupname* lists all of the users that belong to that specific group.

To delete the local CIFS groups

- ◆ To delete the local CIFS group, enter the following:

```
CIFS> local group delete groupname
```

where *groupname* is the name of the local CIFS group.

Enabling CIFS data migration

Veritas Access provides the following command for enabling CIFS data migration:

```
CIFS> set data_migration yes|no
```

To enable data migration for the CIFS server

- 1 To enable data migration for the CIFS server, enter the following:

```
CIFS> set data_migration yes
```

- 2 Restart the CIFS server by entering the following command:

```
CIFS> server start
```

- 3 Map the CIFS share on the Windows domain using the *isa_Cluster_Name\root* by the Domain Administrator.

- 4 Copy the data with ROBOCOPY by entering the following command in a Windows command prompt:

```
C:\> ROBOCOPY /E /ZB /COPY:DATSO [windows_source_dir] [CIFS_target_dir]
```

Make sure you have the Windows Resource Kit Tools installed.

- 5 Disable the CIFS data migration option after migration completes for CIFS server security by entering the following command:

```
CIFS> set data_migration no
```

- 6 Restart the CIFS server by entering the following command:

```
CIFS> server start
```

Configuring Veritas Access to work with Oracle Direct NFS

This chapter includes the following topics:

- [About using Veritas Access with Oracle Direct NFS](#)
- [Best practices for improving Oracle database performance](#)
- [About the Oracle Direct NFS architecture](#)
- [About Oracle Direct NFS node or storage connection failures](#)
- [Configuring an Oracle Direct NFS storage pool](#)
- [Configuring an Oracle Direct NFS file system](#)
- [Configuring an Oracle Direct NFS share](#)

About using Veritas Access with Oracle Direct NFS

Veritas Access lets you create and manage storage for Oracle database clients. Oracle hosts access the storage using Oracle Direct NFS (DNFS).

Oracle Direct NFS is an optimized NFS (Network File System) client that provides faster access to NFS storage that is located on NAS storage devices. The Oracle Database Direct NFS client integrates the NFS client functionality directly in the Oracle software. Through this integration, the I/O path between Oracle and the NFS server is optimized, providing significantly better performance. In addition, the Oracle

Direct NFS client simplifies and, in many cases, automates the performance optimization of the NFS client configuration for database workloads.

The Oracle Direct NFS client outperforms traditional NFS clients, and is easy to configure. The Oracle Direct NFS client provides a standard NFS client implementation across all hardware and operating system platforms.

Veritas Access creates different storage pools for different Oracle object types. Veritas Access has the following storage pools as described in [Table 11-1](#).

Table 11-1 Veritas Access storage pools for Oracle object types

Pool Name	Database Object Type	Function
<code>ora_data_pool</code>	Oracle TABLE data files	Stores TABLE data of data files. This database object type uses striped volumes over four LUNs. The stripe size is 256K.
<code>ora_index_pool</code>	Oracle INDEX files	Stores INDEX data.
<code>ora_temp_pool</code>	Temporary files	Stores temporary files. The storage administrator should make sure the LUNs are from the fastest tier. Temporary files are used for sort, merge, or join queries.
<code>ora_archive_pool</code>	Archive logs	Stores archive logs. This database object type is a concatenated volume. Tier 2 LUNs can be used for this storage pool.
<code>ora_txnlog_pool</code>	REDO txnlog files	Stores REDO transaction logs. It is recommended to assign the fastest storage LUNs to this storage pool.

Best practices for improving Oracle database performance

Oracle database performance depends on I/O bandwidth and latency.

The Oracle database has the following object types:

- DATA
- INDEX
- TXNLOG
- ARCHLOG
- TEMPFILES

See [“About using Veritas Access with Oracle Direct NFS”](#) on page 166.

The Oracle database transaction rate depends heavily on TXNLOG write latency.

Table 11-2 Best practices for improving Oracle database performance

Item	Action
DATA and INDEX files	Separate DATA and INDEX files into separate disk pools with a minimum of four LUNs.
REDOLOG files	Configure REDOLOG files in a separate file system (NFS share). The underlying storage LUNs should not be shared with other volumes or file systems.
TEMPFILES	For data warehouse applications, keep TEMPFILES in separate pools with a minimum of four LUNs.
TXNLOG files	Place TXNLOG files in a separate disk pool with dedicated fast LUNs.
NFS daemon threads	To get better performance, increase NFS daemon threads to 128 or more. If the network permits, you may want to enable jumbo frames. See “About the Oracle Direct NFS architecture” on page 169.
Veritas Access database-specific file systems	Access Veritas Access database-specific file systems from the Oracle host using dedicated virtual IPs. Do not use the same virtual IP for other applications.
Veritas Access file systems	Always use Oracle recommended mount options to mount Veritas Access file systems on the Oracle host. This mount option depends on the Oracle database version. See the <i>Veritas Access Installation Guide</i> for the supported Oracle database versions. See the Oracle documentation for the recommended mount options.

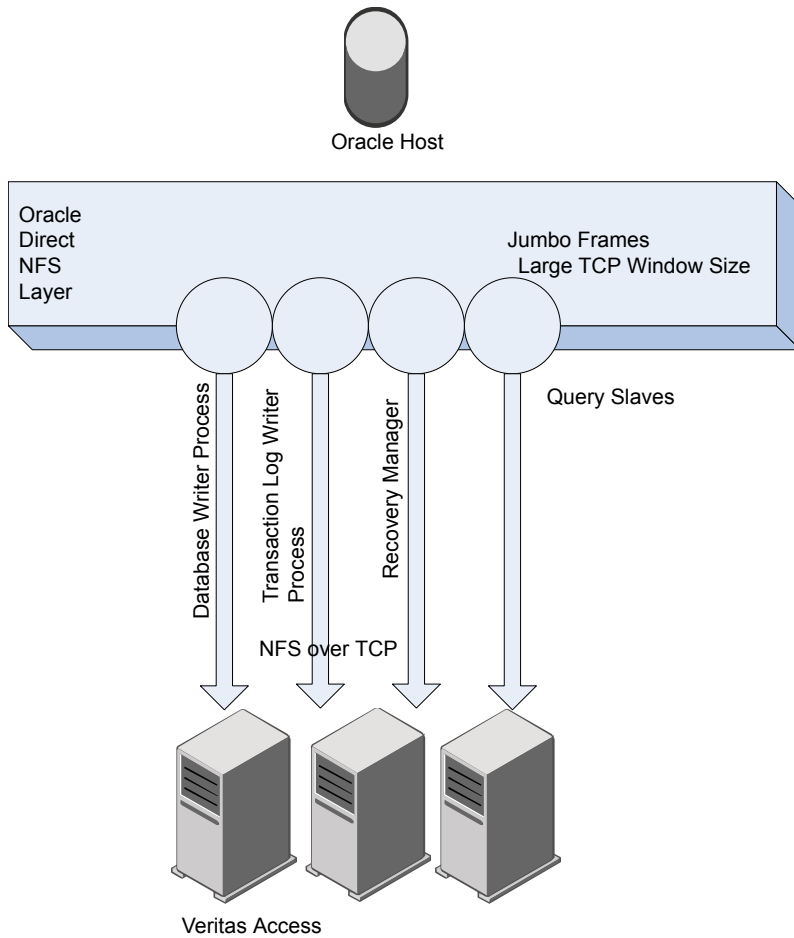
About the Oracle Direct NFS architecture

Oracle Direct NFS can issue 1000s of concurrent operations due to the parallel architecture. Every Oracle process has its own TCP connection.

See the *Veritas Access Installation Guide* for the supported Oracle operating systems.

Figure 11-1 describes the data flow from the Oracle host to Veritas Access.

Figure 11-1 Oracle Direct NFS architecture



Setting a larger frame size on an interface is commonly referred to as using jumbo frames. Jumbo frames help reduce fragmentation as data is sent over the network and in some cases, can also provide better throughput and reduced CPU usage.

You can configure jumbo frames for Oracle Direct NFS by setting the Maximum Transmission Unit (MTU) value.

About Oracle Direct NFS node or storage connection failures

When a node or a storage connection fails, Veritas Access fails over the virtual IP (VIP) to the healthy node. If Oracle does active transactions, some I/O can fail during this VIP failover time window. Oracle generally issues several asynchronous I/O requests using Oracle Direct NFS in parallel. If Oracle detects some I/O failure waiting for completed I/Os, the result depends on the database file type. If the write fails for the REDO transaction log or the database control files, then the database instance fails. If I/O to a data file fails, then that particular data file is taken offline. The SYSTEM and the UNDO data files are considered critical data files. The Oracle database fails if I/O to these critical data files fail. When the database instance fails, the database administrator should wait until the VIP fails over to the healthy Veritas Access node. When the VIP is online, the database administrator can start up the database and then recover the database.

Configuring an Oracle Direct NFS storage pool

To create an Oracle Direct NFS storage pool

- ◆ Use the `Database> pool create` command to create a pool to store database objects.

```
Database> pool create obj-type disk1[,disk2,...]
```

obj-type

Specifies the Oracle object type.

It is recommended to group the storage according to the database objects that are stored in the file system. Oracle database objects are broadly divided into REDO transaction logs, archived logs, table data, index, or temporary files.

Available values include:

- txnlog
- data
- index
- temp
- archivelog

See [“About using Veritas Access with Oracle Direct NFS”](#) on page 166.

disk1, disk2

Specifies the disks to include in the Oracle Direct NFS storage pool.

An error message displays if a disk is not present, or if the disk is already used.

To destroy an Oracle Direct NFS storage pool

- ◆ To destroy a specified Oracle Direct NFS storage pool, enter the following:

```
Database> pool destroy obj-type
```

To list all your Oracle Direct NFS storage pools

- ◆ Use the `Database> pool list` command to list all your Oracle Direct NFS storage pools that are configured for the database.

```
Database> pool list
```

Configuring an Oracle Direct NFS file system

To create an Oracle Direct NFS file system

- ◆ Use the `Database> fs create` command to create an Oracle Direct NFS file system for storing objects.

```
Database> fs create obj-type db_name fs_name size
```

<i>obj-type</i>	<p>Specifies the Oracle object type.</p> <p>Available values include:</p> <ul style="list-style-type: none"> ■ txnlog ■ data ■ index ■ temp ■ archivelog <p>See “About using Veritas Access with Oracle Direct NFS” on page 166.</p>
<i>db_name</i>	Specifies the Oracle database name.
<i>fs_name</i>	<p>Specifies the file system name for which you want to create.</p> <p>For a given database, Veritas recommends having at least three file systems provisioned from respective storage pools:</p> <ul style="list-style-type: none"> ■ One file system for txnlog ■ One file system for data ■ One file system for archivelog
<i>size</i>	Specifies the size of the file system that you want to create.

To destroy an Oracle Direct NFS file system

- ◆ Use the `Database> fs destroy` command to destroy an Oracle Direct NFS file system.

```
Database> fs destroy db_name fs_name
```

To list the Oracle Direct NFS file systems

- ◆ Use the `Database> fs list` command to list the Oracle Direct NFS file systems that are created for storing database files.

```
Database> fs list
```

Configuring an Oracle Direct NFS share

To configure an Oracle Direct NFS share

- 1 Use the `Database> share add` command to share and export a file system. After issuing the `Database> share add` command, database clients are able to NFS mount the specified file system on their Oracle host.

```
Database> share add obj-type export_dir [client]
```

obj_type

Specifies the Oracle object type.

Available values include:

- txnlog
- data
- index
- temp
- archivelog

See [“About using Veritas Access with Oracle Direct NFS”](#) on page 166.

export_dir

Specifies the directory location of the exported file system.

client

Specifies the database client.

- 2 Use the `Database> share show` command to display all the shared database file systems.

```
Database> share show
```

- 3 Use the `Database> share delete` command to delete or unshare the exported file system.

```
Database> share delete export-dir [client]
```

Configuring an FTP server

This chapter includes the following topics:

- [About FTP](#)
- [Creating the FTP home directory](#)
- [Using the FTP server commands](#)
- [About FTP server options](#)
- [Customizing the FTP server options](#)
- [Administering the FTP sessions](#)
- [Uploading the FTP logs](#)
- [Administering the FTP local user accounts](#)
- [About the settings for the FTP local user accounts](#)
- [Configuring settings for the FTP local user accounts](#)
- [File sharing for a scale-out file system using FTP](#)

About FTP

The file transfer protocol (FTP) server feature allows clients to access files on the Veritas Access servers using the FTP protocol. The FTP service provides secure/non-secure access by FTP to files in the Veritas Access servers. The FTP service runs on all of the nodes in the cluster and provides simultaneous read and write access to the files. The FTP service also provides configurable for anonymous access to Veritas Access.

By default, the FTP server is not running. You can start the FTP server using the `FTP> server start` command. The FTP server starts on the standard FTP port 21.

The Veritas Access FTP service does not support transparent failover. During failover due to either a shutdown or a restart of the server, the FTP client loses its connection to the server. As a consequence, any upload or download to the FTP service during the failover fails. Restart any upload or download to the FTP service from the beginning after the connection to the FTP service has been re-established.

Creating the FTP home directory

Veritas Access can act as an FTP server for LDAP, NIS, or AD users, or local users.

When a user logs into the FTP server for the first time, Veritas Access retrieves the user's home directory information from the authentication server. The authentication server can be an LDAP, NIS, or AD server.

If the `create_homedirs` option is set to `yes`, Veritas Access creates a user's home directory on the FTP server with the same name that was retrieved from the authentication server. This directory is used internally. If the `create_homedirs` option is set to `no`, the Veritas Access administrator must manually create a directory that matches the home directory on the authentication server.

Regardless of the setting of the `create_homedirs` option, the Veritas Access administrator must manually create the user's directory where the user logs in. This directory is in the location specified by the `homedir_path` option. The directory must have execute permissions set.

Using the FTP server commands

The `FTP> server` commands start, stop, and display the status of the FTP server.

To display the FTP server status

- ◆ To display the FTP server status, enter

```
FTP> server status
```

To start the FTP server

- 1 If the attribute `user_logon` is set to `yes` (the default value), set a value for `homedir_path`.

The `homedir_path` must be set before the FTP server can start.

```
FTP> set homedir_path pathname
```

Where:

<i>pathname</i>	Specifies the location of the login directory for users. Valid values include any path that starts with <code>/vx/</code> .
-----------------	-----------------------------------------------------------------------------------------------------------------------------

- 2 To start the FTP server, enter the following:

```
FTP> server start
```

To check server status, enter the following:

```
FTP> server status
```

To stop the FTP server

- ◆ To stop the FTP server, enter the following:

```
FTP> server stop
```

To check the server status, enter the following:

```
FTP> server status
```

About FTP server options

Veritas Access lets you set various configurable options for the FTP server.

For the changes to take effect, restart the FTP server.

Table 12-1 FTP options

Option	Definition
<code>allow_delete</code>	<p>Specifies whether or not to allow users to delete files on the FTP server. This option only applies to users. It does not apply to anonymous logins. Anonymous logins are never allowed to delete files.</p> <p>Enter <i>yes</i> (default) to allow users to delete files on the FTP server. Enter <i>no</i> to prevent users from deleting files on the FTP server.</p>
<code>allow_non_ssl</code>	<p>Specifies whether or not to allow non-secure (plain-text) logins into the FTP server. Enter <i>yes</i> (default) to allow non-secure (plain-text) logins to succeed. Enter <i>no</i> to allow non-secure (plain-text) logins to fail.</p>
<code>anonymous_login_dir</code>	<p>Specifies the login directory for anonymous users. Valid values of this parameter start with <i>/vx/</i>. Make sure that the anonymous user (UID:40 GID:49 UNAME:ftp) has the appropriate permissions to read files in <i>login_directory</i>.</p>
<code>anonymous_logon</code>	<p>Tells the FTP server whether or not to allow anonymous logons. Enter <i>yes</i> to allow anonymous users to log on to the FTP server. Enter <i>no</i> (default) to not allow anonymous logons.</p>
<code>anonymous_write</code>	<p>Specifies whether or not anonymous users have the [write] value in their <i>login_directory</i>. Enter <i>yes</i> to allow anonymous users to modify contents of their <i>login_directory</i>. Enter <i>no</i> (default) to not allow anonymous users to modify the contents of their <i>login_directory</i>. Make sure that the anonymous user (UID:40 GID:49 UNAME:ftp) has the appropriate permissions to modify files in their <i>login_directory</i>.</p>
<code>chroot_users</code>	<p>Specifies whether users should be restricted to their home directories. A value of <i>yes</i> limits users to their home directory. A value of <i>no</i> allows users to view files in parent directories. Users are restricted by their <i>homedir_path</i>. If security is local, then <i>chroot_users</i> should be set to <i>yes</i>.</p>
<code>create_homedirs</code>	<p>Specifies if home directories should be created when a user logs in, if the home directory does not exist. A value of <i>yes</i> allows FTP to create a user's home directory, if it does not already exist. If the value is <i>no</i>, then a home directory should exist for this user, and the user should have permissions to read and execute in this directory. Otherwise, the login fails.</p>

Table 12-1 FTP options (*continued*)

Option	Definition
<code>homedir_path</code>	Specifies the location of the login directory for users. Valid values include any path that starts with <code>/vx/</code> . This option is required if <code>user_logon</code> is set to <code>yes</code> .
<code>idle_timeout</code>	Specifies the amount of time in minutes after which an idle connection is disconnected. Valid values for <i>time_in_minutes</i> range from 1 to 600 (default value is 15 minutes).
<code>listen_ipv6</code>	Specifies whether the FTP service should listen on IPv6 for connections. Valid values for this parameter are <code>yes</code> or <code>no</code> . The default value is <code>no</code> .
<code>listen_port</code>	Specifies the port number on which the FTP service listens for connections. Valid values for this parameter range from 10-1023. The default value is 21.
<code>max_connections</code>	Specifies the maximum number of simultaneous FTP clients allowed. Valid values for this parameter range from 1-9999. The default value is 2000.
<code>max_conn_per_client</code>	Specifies the maximum number of simultaneous FTP connections that are allowed from a single client IP address. Valid values for this parameter range from 1-9999. The default value is 2000.
<code>passive_port_range</code>	Specifies the range of port numbers to listen on for passive FTP transfers. The <i>port_range</i> defines a range that is specified as <code>startingport:endingport</code> . A <i>port_range</i> of <code>30000:40000</code> specifies that port numbers starting from 30000 to 40000 can be used for passive FTP. Valid values for port numbers range from 30000 to 50000. The default value of this option is <code>30000:40000</code> .
<code>security</code>	<p>Specifies the type of users that are allowed to log in to the FTP server. Enter <i>nls_lap</i> (default) to allow users with accounts configured on NIS or LDAP servers to log in to the FTP server. Users that are created with the <code>FTP > local user add</code> command cannot log in.</p> <p>Enter <i>local</i> to allow users with accounts created with the <code>FTP> local user add</code> command to log in to the FTP server. NIS and LDAP users cannot log in.</p> <p>The <i>ads</i> option allows access to users configured on Windows Active Directory as specified in the <code>CIFS> show</code> command. NIS, LDAP, and local users are not allowed to log in.</p>

Table 12-1 FTP options (*continued*)

Option	Definition
<code>umask</code>	<p>Specifies the mask for permissions with which files or directories are created using FTP.</p> <p>If the <code>file_umask</code> is set to 177, then new files and directories are created with permissions 600, which defines <code>rw-----</code>. The owner of the file or directory has read and write permissions to the file or directory. Members in the users group do not have read or write permissions.</p>
<code>user_logon</code>	<p>Specifies whether to allow FTP access for users. A value of <code>yes</code> allows normal users (non-anonymous users) to log in.</p> <p>If <code>user_logon</code> is set to <code>yes</code>, then the <code>homedir_path</code> also must be set or the FTP server cannot start.</p>

Customizing the FTP server options

The `FTP> set` commands let you set various configurable options for the FTP server.

For the changes to take effect, the FTP server must be restarted.

To change the FTP server options

- 1 To view the current settings or view the pending command changes , enter the following:

```
FTP> show
```

- 2 To change the required server options, use the `set` command.

For example, to enable anonymous logons, enter the following:

```
FTP> set anonymous_logon yes
```

- 3 To implement the changes, you must stop and restart the FTP server.

Enter the following:

```
FTP> server stop
FTP> server start
```

- 4 To view the new settings, enter the following:

```
FTP> show
```

Administering the FTP sessions

To display the current FTP sessions

- ◆ To display the current FTP sessions, enter the following:

```
FTP> session show
```

To display the FTP session details

- ◆ To display the details in the FTP sessions, enter the following:

```
FTP> session showdetail [filter_options]
```

where *filter_options* display the details of the sessions under specific headings. Filter options can be combined by using ','. If multiple filter options are used, sessions matching all of the filter options are displayed.

To display all of the session details, enter the following:

```
FTP> session showdetail
```

To terminate an FTP session

- ◆ To terminate one of the FTP sessions that are displayed in the `FTP> session showdetail` command, enter the following:

```
FTP> session terminate session_id
```

where *session_id* is the unique identifier for each FTP session that is displayed in the `FTP> session showdetail` output.

Uploading the FTP logs

The `FTP> logupload` command lets you upload the FTP server logs to a specified URL.

To upload the FTP server logs

- ◆ To upload the FTP server logs to a specified URL, enter the following:

```
FTP> logupload url [nodename]
```

url	<p>The URL where the FTP logs are uploaded. The URL supports both FTP and SCP (secure copy protocol). If a node name is specified, only the logs from that node are uploaded.</p> <p>The default name for the uploaded file is <code>ftp_log.tar.gz</code>.</p> <p>Passwords that are added directly to the URL are not supported.</p>
nodename	<p>The node on which the operation occurs. Enter the value <code>all</code> for the operation to occur on all of the nodes in the cluster.</p>
password	<p>Use the password you already set up on the node to which you upload the logs.</p>

Administering the FTP local user accounts

The `FTP> local user` commands let you create and manage local user accounts on the FTP server.

When you add a local user account, the user's home directory is created automatically on the FTP server. User home directories on the FTP server are specified by *path/username* where *path* is the home directory path configured by the `FTP > set homedir_path` command.

All users are limited to their home directories and are not allowed to access files on the FTP server beyond their home directories.

To add a local user account

- 1 To add a local user account, enter the following:

```
FTP> local user add username
```

where *username* is the name of the user whose account you want to add.

- 2 When the password prompt appears, enter a password for the local user.
- 3 Type the password again for verification.

To change a password for a local user

- 1 To change a password for a local user, enter the following:

```
FTP> local user passwd username
```

where *username* is the name of the user whose password you want to change.

- 2 When the password prompt appears, enter a new password, then type the password again for verification.

To delete a local user account

- ◆ To delete a local user account, enter the following:

```
FTP> local user delete username
```

where *username* is the name of the user whose account you want to delete.

When you delete a local user account, the local user's home directory is not deleted.

To show local user accounts

- ◆ To show local user accounts (and account settings) configured on the FTP server, enter the following:

```
FTP> local user show
```

About the settings for the FTP local user accounts

By default, local user accounts on the FTP server have no limits for the following:

- Bandwidth.
- Number of simultaneous connections.

To configure limits for these options, use the `FTP> user local set` commands.

You can also use the `FTP> local user set` command to specify home directories for local users accounts.

Local user changes are effective immediately for new connections. You do not need to restart the FTP server.

Table 12-2 FTP local user options

Option	Definition
bandwidth	Specifies the maximum bandwidth (in MB/second) for a local user account on the FTP server. By default, there is no limit on the bandwidth for local users.
max_connections	Specifies the maximum number of simultaneous connections a local user can have to each node in the cluster. By default there is no limit to the number of connections a local user can have to the FTP server.
homedir	<p>Specifies the home directory for a local user account.</p> <p>The home directory you configure for a local user account is created relative to the home directory path that is configured by the <code>FTP > set homedir_path</code> command.</p> <p>The default home directory value for local user accounts is <i>username</i> where <i>username</i> is the login name for the local user account.</p> <p>For example, if the home directory path is set to <code>/vx/fsl/ftp_home</code> and the user name is <code>user1</code>, the default home directory for <code>user1</code> is <code>/vx/fsl/ftp_home/user1</code></p> <p>Changes to this value are applicable for any new connections. Configuring a new home directory location does not migrate any existing data in a local user's current home directory to the new home directory.</p>

Configuring settings for the FTP local user accounts

To show local user settings

- ◆ To show the current settings for local user accounts, enter the following:

```
FTP> local user show
```

To set bandwidth

- ◆ To set the maximum bandwidth for a local user account, enter the following:

```
FTP> local user set bandwidth username max_value
```

username Specifies the name of a user account.

max_value Specifies the maximum upload bandwidth value (measured in MB/second) for the user's account.

To set maximum connections

- ◆ To set the maximum number of simultaneous connections a local user can have to the FTP server, enter the following:

```
FTP> local user set max_connections username
number
```

username Specifies the name of a user account.

number Specifies the maximum number of simultaneous connects a user can have to the FTP server.

To set the home directory

- ◆ To set the home directory for a local user account, enter the following:

```
FTP> local user set homedir username
dir_name
```

username Specifies that name of a user account.

dir_name Specifies the name of the home directory for the local user account.

The home directory you configure for a local user account is relative to the home directory path that is configured by the `FTP> set homedir_path` command.

Changes to this value are applicable for any new connections. Configuring a new home directory location does not migrate any existing data in a local user's current home directory to the new home directory.

File sharing for a scale-out file system using FTP

Veritas Access provides support for sharing files using the FTP protocol for a scale-out file system. A scale-out file system can scale linearly, thus providing an advantage over a standard clustered file system.

See [“About scale-out file systems”](#) on page 211.

The following limitations apply when you share files using the FTP protocol for a scale-out file:

- Quota settings
The `FTP> homedir set` command is usually used along with the `Storage> quota` functionality. Since the scale-out file system does not support quota, an FTP home directory which is based on a scale-out file system does not have quota functionality.
- FTP with LDAP as security is not supported.

Managing the Veritas Access Object Store server

- [Chapter 13. Using Veritas Access as an Object Store server](#)

Using Veritas Access as an Object Store server

This chapter includes the following topics:

- [About the Object Store server](#)
- [Use cases for configuring the Object Store server](#)
- [Configuring the Object Store server](#)
- [About buckets and objects](#)
- [File systems used for objectstore buckets](#)
- [Multi-protocol support for NFS with S3](#)

About the Object Store server

The Veritas Access Object Store server lets you store and retrieve the data that is stored in Veritas Access using the Amazon Simple Storage Service (S3) compatible protocol. The protocol implements the RESTful API interface over standard HTTP protocol.

See the ObjectAccess service (S3) APIs section in the *Veritas Access RESTful API Guide* for more information on the APIs supported by the Veritas Access Object Store server.

Features of the Object Store server:

- High availability
- Customization of storage layouts as per requirement
- Concurrent access from multiple nodes

- Scalability with multiple nodes and buckets
- Sharing of file systems and customization of file system layout using groups

Use cases for configuring the Object Store server

You can configure the Object Store server depending on different use cases.

Use Case 1: Large number of objects per bucket are required.

- The admin can configure a default pool without using the `fs_sharing` option.
- The file system is not shared across buckets. A bucket can have large number of objects. The choice of file system sharing limits the number of buckets created.

Use Case 2: Admin needs large number of buckets but does not expect large number of objects per bucket.

- The admin can create a group in its authentication server and configure this group in Object Store using the `objectaccess> group set` command.
- The grouping provides options like choosing the disk pool to use, file system type, file system sharing, file system size, other file system options.
- The admin can use the `fs_sharing` option to configure the Object Store server to share a file system across all buckets that are created by a user of that particular group.
- The file system sharing allows the Object Store server to create a large number of buckets but limits the total number of objects present across the bucket.

Use Case 3: Admin wants to control the file system used for a bucket.

- The admin has to pre-create the required file system using the `storage> fs` commands.
- The admin can use the `objectaccess> map` command to map a directory of the existing file system as a bucket for a particular user.

Configuring the Object Store server

To configure the Object Store server

- 1 Log on to Veritas Access using the CLISH.
- 2 Create a default storage pool (at least one) on the cluster.

```
CLISH> storage pool create pool1 disk1,disk2,disk3,disk4
```

- 3 Use the storage pool that was created in Step 2 as the default object access pool.

You need to set the default pool, as it is required for enabling the Object Store server.

```
CLISH> objectaccess set pools pool1
```

Note: Based on your storage requirements, you can configure different types of storage pools by using the Object Store group commands.

- 4 Verify the configured storage pool.

```
CLISH> objectaccess show
```

- 5 Enable and start the Object Store server.

```
CLISH> objectaccess server enable
```

```
CLISH> objectaccess server start
```

- 6 Configure the cluster using any authentication server (AD, LDAP, or NIS).

See the following manual pages for more information on configuring AD, LDAP, or NIS:

- `CLISH> network man ldap`
- `CLISH> man cifs`
- `CLISH> man nis`

- 7 Create the access and secret keys for the authorized user, or any user in the authentication server.

You have two options for creating the access and the secret keys, either using the Veritas Access RESTful APIs or by using the Veritas Access helper script.

Create the access and secret keys using the Veritas Access RESTful APIs:

- Before using the Veritas Access RESTful APIs, set the host name resolution for the host as shown in the `objectaccess> show` output against `ADMIN_URL`.
- See the *Veritas Access RESTful API Guide* on the [SORT](#) site for accessing the Object Store server (S3) user management APIs.

- After creating your access and secret key, you can create a bucket using the S3 API.

Create the access and the secret keys using the Veritas Access helper script:

- Add the `ADMIN_URL` name in your `/etc/hosts` file.
where the `ADMIN_URL` is `admin.<cluster_name>` and the port is 8144. This url should point to the Veritas Access management console IP address.
- Location of the helper script:
`/opt/VRTSnas/scripts/utils/objectaccess/objectaccess_client.py`
- The Veritas Access helper script can be used from any client system that has Python installed.
- To run the script, your S3 client needs to have the `argparse` and `requests` Python modules.
If these modules are missing, install both these modules using `pip` or `easy_install`.
- Create the access and the secret key using the Veritas Access helper script by providing the user name, password, and `ADMIN_URL` (check the online Help of the Veritas Access helper script for all of the provided operations like `list key` and `delete key`).

Create a secret key:

```
clus_01:~ # ./objectaccess_client.py --create_key
--server admin.clus:8144 --username localuser1 --password root123
--insecure
UserName                : localuser1
AccessKeyId              : Y2FkODU2NTU2MjVhYzV
Status                   : Active
SecretAccessKey          : ODk0YzQxMDhkMmRjM2M5OTUzNjI5OWIzMdgyNzY
```

The `<localuser1>` is the local user created on both the Veritas Access cluster nodes with same unique ID.

List a secret key for the specified user:

```
clus_01:~ # ./objectaccess_client.py --list_key --server
admin.clus:8144 --username localuser2 --password root123 --insecure
```

Delete a secret key for the specified user:

```
clus_01:~ # ./objectaccess_client.py --delete_key
ZTkYNDdjZTViM2EyMWZ --server admin.clus:8144 --username localuser2
--password root123 --insecure
```

- If the Object Store server is enabled without the `SSL` option, you need to add the `--insecure` option.

```
clus_01 ~# ./objectaccess_client.py --server
admin.clus:8144 --username <uname> --create_key --insecure
```

- 8 Use the following `objectaccess` command to see all the existing access and secret keys in the Veritas Access cluster:

```
CLISH> objectaccess account user show
```

Changing the Object Store server options

It is possible to change an already set parameter or set new parameters by specifying different options. For example, you can change the other Object Store server defaults, such as `fs_type`, `fs_size`, and other options.

After setting the defaults, you can verify whether the proper value is assigned or not.

```
vmdellr> objectaccess set fs_type
ecoded largefs mirrored mirrored-stripe simple striped striped-mirror
```

```
vmdellr> objectaccess set fs_type simple
ACCESS ObjectAccess INFO V-288-0 Set fs_type successful.
```

```
vmdellr> objectaccess set fs_size 2G
ACCESS ObjectAccess INFO V-288-0 Set operation successful.
```

```
vmdellr> objectaccess show
```

Name	Value
Server Status	Enabled
Admin_URL	http://admin.vmdellr:8144
S3_URL	http://s3.vmdellr:8143
admin_port	8144
s3_port	8143
ssl	no
poollist	['pool1']
fs_size	2G
fs_blksize	8192
fs_pdirenable	no
fs_encrypt	off
fs_type	simple

Using the group option for bucket creation

If you have multiple users, and you want to set different default values for different sets of users, you can use the `group` option.

You can also use the `group` option to use the existing file systems for bucket creation instead of creating a new file system for every bucket. If you set the `group fs_sharing` option to **yes**, and if any request for bucket creation comes from a user who is part of that group, then the S3 server searches for any existing file system created by the specific group user. If an existing file system is found, it uses the existing file system. Otherwise, it creates a new file system for the bucket.

To use the group option

- 1 Create a group in the authentication server (AD/LDAP/NIS) and add the required users to that group.
- 2 Set the group specific configuration for the group created in the authentication server.
- 3 Set or unset the defaults per your requirements.

```
vmdellr> objectaccess group set fs_type simple VRTS-grp
ACCESS ObjectAccess INFO V-288-0 Group set fs-type successful.
```

```
vmdellr> objectaccess group set pool VRTS-grp pool1
ACCESS ObjectAccess INFO V-288-0 Success.
```

```
vmdellr> objectaccess group show
```

Group Name	Fs Sharing	Fs Size	Fs Type	Pool(s)
VRTS-grp	-	-	simple	pool1

```
vmdellr> objectaccess group show
```

Group Name	Fs Sharing	Fs Size	Fs Type	Pool(s)
VRTS-grp	-	-	-	pool1

About buckets and objects

The Object Store sever consists of a collection of objects. The container of an object is known as a bucket. In Veritas Access Object Store, the buckets are stored on file systems as directories and object are stored as files.

Buckets and objects are resources which can be managed using the APIs.

Once the Object Store Server is configured, you can create bucket and objects and perform the required operations.

Veritas Access supports the following methods for accessing the buckets and the objects:

- Path-style method
- Virtual-hosted-style method
When using the virtual hosted-style method, the ***bucket_name.s3.cluster_name*** should be DNS resolvable.
See the `objectaccess_bucket(1)` manual page for more information.

See the `objectaccess` manual pages for all of the Veritas Access Object Store server operations that can be performed.

Buckets are created by S3 clients by calling the standard S3 APIs to the Veritas Access S3 server. For creating a bucket, you need the endpoint of the Veritas Access server, access key, and the secret key. The endpoint of the Veritas Access Object Store server is `s3.cluster_name:8143`.

The Veritas Access Object Store server can also be accessed using the fully qualified domain name:

s3.cluster_name.fqdn:8143

Make sure that you associate one (or more) of the VIPs of the Veritas Access cluster to `s3.cluster_name.fqdn` in the client's DNS server.

[Table 13-1](#) describes the restrictions enforced by the Veritas Access Object Storage Server. Configure your S3 clients within these limitations to ensure that Veritas Access works correctly.

Table 13-1 Object and bucket restrictions

Description	Limit
Maximum recommended parallel threads	10
Maximum number of buckets per file system with <code>fs_sharing</code> enabled	10,000
Maximum number of objects per file system	1 billion
Maximum supported size of an object that can be uploaded using a single PUT	100 MB
Maximum number of parts supported for multipart upload	10,000
Maximum supported size range of an object that can be downloaded using a single GET	100 MB

Table 13-1 Object and bucket restrictions (*continued*)

Description	Limit
Maximum number of grantees supported for setting ACL on buckets/objects	128

File systems used for objectstore buckets

Veritas Access supports the following file systems for creating buckets:

- Ecoded
- Largefs
- Mirrored
- Mirrored-stripe
- Simple
- Striped
- Striped-mirror

Note: The default file system layout for Veritas Access 3340 Appliance is `largefs`.

Multi-protocol support for NFS with S3

Veritas Access supports multi-protocol support for NFS with S3. If an NFS share is present (and objects may be present in the exported path), the storage admin can map that path as an S3 bucket (S3 over NFS). In addition, a normal file system path can also be mapped as an S3 bucket. The buckets created by S3 APIs cannot be exported as an NFS share (NFS over S3).

Obtaining the path to map as S3 bucket

The path has the following characteristics:

- Path is the absolute path inside a file system.
- The name of the bucket is the name of the directory of the path which should be S3 compliant.
- The path can be either NFS exported path or any directory in the normal file system. You cannot not use the ObjectAccess file systems (file system having S3 bucket created by S3 APIs).
- No other bucket should exist with the same name.

- No other bucket should be present either inside or outside the given path. You can verify this using the following command:

```
objectaccess> bucket show
```

- NFS share should not be present before or after that directory. You can verify using the following command:

```
NFS> share show
```

Creating an S3 user

You can configure the cluster with any authentication server like AD/LDAP/NIS. Then, all the users present in the authentication server can be used as S3 users.

The S3 user should be authorized to access the S3 bucket (access key and secret key should be present for that user). You can verify using the following command:

```
objectaccess> account user show
```

See [“Configuring the Object Store server”](#) on page 188.

Mapping the path to the S3 bucket for the user

You can map the path to the S3 bucket for the user using the following command:

```
objectaccess> map <path> <user>
```

The storage admin can verify the bucket creation using the following command:

```
objectaccess> bucket show
```

Using the multi-protocol feature

The storage admin can use the NFS share at the same time when the S3 user uses the bucket. Existing objects inside the bucket retain the permissions set by the owner of those objects.

Unmapping the S3 bucket

In multi-protocol case, an S3 user can delete bucket without deleting all the objects. Deleting the bucket is equivalent to unmapping or unreferencing the bucket.

Limitations

The following limitations apply for multi-protocol support:

- An S3 user cannot access a bucket if the bucket ownership or permissions from the NFS client is changed.

- Permissions that are set or modified from protocols like NFS are not honored by S3 and vice versa.
- Object ETag is inaccurate whenever object is created or modified from the NFS client. An incorrect ETag is corrected when a GET or HEAD request is performed on the object.
- Accessing the same object from different protocol in exclusive mode is not supported.

Monitoring and troubleshooting

- [Chapter 14. Configuring event notifications and audit logs topics](#)

Configuring event notifications and audit logs topics

This chapter includes the following topics:

- [About event notifications](#)
- [About severity levels and filters](#)
- [About SNMP notifications](#)
- [Configuring an email group](#)
- [Configuring a syslog server](#)
- [Exporting events in syslog format to a given URL](#)
- [Displaying events on the console](#)
- [Configuring events for event reporting](#)
- [Configuring an SNMP management server](#)

About event notifications

Veritas Access monitors the status and health of various network and storage components, and generates events to notify the administrator. Veritas Access provides a mechanism to send these events to external event monitoring applications like syslog server, SNMP trap logger, and mail servers. This section explains how to configure Veritas Access so that external event monitoring applications are notified of events on the Veritas Access cluster.

About severity levels and filters

Veritas Access monitors events of different severity levels. Set the severity to a particular level to specify the severity level to include in notifications. Notifications are sent for events having the same or higher severity.

Table 14-1 describes the valid Veritas Access severity levels in descending order of severity.

Table 14-1 Severity levels

Valid value	Description
emerg	Indicates that the system is unusable
alert	Indicates that immediate action is required
crit	Indicates a critical condition
err	Indicates an error condition
warning	Indicates a warning condition
notice	Indicates a normal but a significant condition
info	Indicates an informational message
debug	Indicates a debugging message

Veritas Access also classifies event notifications by type. Set the event filter to specify which type of events to include in notifications. Notifications are sent only for events matching the given filter.

The filter is set to one of the following options:

- Network - for networking events
- Storage - for storage-related events. For example, events related to file systems, snapshots, disks, and pools.
- Replication - for replication-related events.
- All - resets the filter to show all events.

For example, if the filter is set to `network`, a network event triggers a notification. A storage-related event would not trigger a notification.

About SNMP notifications

Simple Network Management Protocol (SNMP) is a network protocol to simplify the management of remote network-attached devices such as servers and routers. SNMP is an open standard system management interface. Information from the Management Information Base (MIB) can also be exported.

SNMP traps enable the reporting of a serious condition to a management station. The management station is then responsible for initiating further interactions with the managed node to determine the nature and extent of the problem.

See [“About severity levels and filters”](#) on page 199.

Configuring an email group

Veritas Access can be configured to send email messages to users or groups of users through an external SMTP server.

To display attributes of an email group

- ◆ To display attributes of an email group, enter the following:

```
Report> email show [group]
```

where *group* is optional, and it specifies the group for which to display the attributes. If the specified group does not exist, an error message is displayed. For example:

To add a new email group

- ◆ To add a new email group, enter the following:

```
Report> email add group group
```

where *group* specifies the name of the new email group and can only contain the following characters:

- Alpha characters
- Numbers
- Hyphens
- Underscores

If the entered group already exists, then no error message is displayed.

Multiple email groups can be defined, each with their own email addresses, event severity, and filter.

To add an email address to an existing group

- ◆ To add an email address to an existing group, enter the following:

```
Report> email add email-address group email-address
```

group	Specifies the group to which the email address is added. The email group must already exist.
email-address	Specifies the email address to be added to the group.

To add a severity level to an existing email group

- ◆ To add a severity level to an existing email group, enter the following:

```
Report> email add severity group severity
```

group	Specifies the email group for which to add the severity. The email group must already exist.
severity	Indicates the severity level to add to the email group. See “About severity levels and filters” on page 199. Only one severity level is allowed at one time. You can have two different groups with the same severity levels and filters. Each group can have its own severity definition. You can define the lowest level of the severity that triggers all other severities higher than it.

To add a filter to an existing group

- ◆ To add a filter to an existing group, enter the following:

```
Report> email add filter group filter
```

group	Specifies the email group for which to apply the filter. The email group must already exist.
filter	Specifies the filter for which to apply to the group. See “About severity levels and filters” on page 199. The default filter is <code>all</code> . A group can have more than one filter, but there may not be any duplicate filters for the group.

To delete an email address from an existing group

- ◆ To delete an email address from an existing group, enter the following:

```
Report> email del email-address group email-address
```

group Specifies the group from which to delete the email address.

email-address Specifies the email address from which to delete from the group.

To delete a filter from an existing group

- ◆ To delete a filter from an existing group, enter the following:

```
Report> email del filter group filter
```

group Specifies the group to remove the filter from.

filter Specifies the filter to be removed from the group.

See [“About severity levels and filters”](#) on page 199.

The default filter is `all`.

To delete an existing email group

- ◆ To delete an existing email group, enter the following:

```
Report> email del group group
```

where *group* specifies the name of the email group to be deleted.

To delete a severity from a specified group

- ◆ To delete a severity from a specified group, enter the following:

```
Report> email del severity group severity
```

group Specifies the name of the email group from which the severity is to be deleted.

severity Specifies the severity to delete from the specified group.

See [“About severity levels and filters”](#) on page 199.

To display mail server settings

- ◆ To display mail server settings, enter the following:

```
Report> email get
```

To add a mail server and user account

- ◆ To add a mail server and user account from which email notifications are sent out, enter the following:

```
Report> email set [email-server] [email-user]
```

email-server	Specifies the external mail server from which email notifications are sent out.
--------------	---------------------------------------------------------------------------------

email-user	Specifies the user account from which email notifications are sent out.
------------	-------------------------------------------------------------------------

If `email-user` is specified, then the password for that user on the SMTP server is required.

To delete the mail server from sending email messages

- ◆ To delete the mail server from sending email messages, enter the following command without any options:

```
Report> email set
```

Configuring a syslog server

Veritas Access can be configured to send syslog messages to syslog servers based on set severities and filters.

In Veritas Access, options include specifying the external system log (syslog) server for event reporting, and setting the filter and the severity levels for events. Event notifications matching configured severity levels and filters are logged to those external syslog servers.

See [“About severity levels and filters”](#) on page 199.

To display the list of syslog servers

- ◆ To display the list of syslog servers, enter the following:

```
Report> syslog show
```

To add a syslog server to receive event notifications

- ◆ To add a syslog server to receive event notifications, enter the following:

```
Report> syslog add syslog-server-ipaddr
```

where *syslog-server-ipaddr* specifies the host name or the IP address of the external syslog server.

To set the severity of syslog messages

- ◆ To set the severity of syslog messages to be sent, enter the following:

```
Report> syslog set severity value
```

where *value* indicates the severity of syslog messages to be sent.

See [“About severity levels and filters”](#) on page 199.

To set the filter level of syslog messages

- ◆ To set the filter level of syslog messages to be sent, enter the following:

```
Report> syslog set filter value
```

where *value* indicates the filter level of syslog messages to be sent.

See [“About severity levels and filters”](#) on page 199.

To display the values of the configured filter and severity level settings

- ◆ To display the values of the configured filter and severity level settings, enter the following:

```
Report> syslog get filter|severity
```

To delete a syslog server from receiving message notifications

- ◆ To delete a syslog server from receiving message notifications, enter the following:

```
Report> syslog delete syslog-server-ipaddr
```

syslog-server-ipaddr specifies the host name or the IP address of the syslog server.

Exporting events in syslog format to a given URL

You can export events in syslog format to a given URL.

Supported URLs for upload include:

- FTP
- SCP

To export events in syslog format

- ◆ To export events in syslog format to a given URL, enter the following:

```
Report> exportevents url
```

url Exports the events in syslog format to the specified URL. URL supports FTP and SCP. If the URL specifies the remote directory, the default file name is `access_event.log`.

Displaying events on the console

To display events on the console

- ◆ To display events on the console, enter the following:

```
Report> showevents [number_of_events]
```

where *number_of_events* specifies the number of events that you want to display. If you leave *number_of_events* blank, or if you enter **0**, Veritas Access displays all of the events in the system.

Configuring events for event reporting

To reduce duplicate events

- ◆ To reduce the number of duplicate events that are sent for notifications, enter the following:

```
Report> event set dup-frequency number
```

where *number* indicates time (in seconds) in which only one event (of duplicate events) is sent for notifications.

where *number* indicates the number of duplicate events to ignore.

```
Report> event set dup-number number
```

To display the time interval or the number of duplicate events sent for notifications

- ◆ To display the time interval, enter the following:

```
Report> event get dup-frequency
```

To set the number of duplicate events that are sent for notifications, enter the following:

```
Report> event get dup-number
```

To set the time interval for scanning event notifications

- ◆ To set the time interval for scanning event notifications in `/var/log/messages` and `/var/log/messages-*.bz2` files, enter the following:

```
Report> event set log-scan-frequency frequency
```

where *frequency* is the time interval in seconds for scanning the `/var/log/messages` directory.

To display the time interval for scanning event notifications

- ◆ To display the time interval for scanning event notifications, enter the following:

```
Report> event get log-scan frequency
```

To set the from email address when sending email notifications to users

- ◆ To set the from email address when sending email notifications to users, enter the following:

```
Report> event set from-address from-email-address
```

where *from-email-address* is the from email address when sending email notifications to users.

To display the from email address when sending email notifications to users

- ◆ To display the from email address when sending email notifications to users, enter the following:

```
Report> event get from-address
```

Configuring an SNMP management server

To add an SNMP management server to receive SNMP traps

- ◆ To add an SNMP management server to receive SNMP traps, enter the following:

```
Report> snmp add snmp-mgmtserver-ipaddr [community_string]
```

snmp-mgmtserver-ipaddr specifies the host name or the IP address of the SNMP management server.

[*community_string*] specifies the community name for the SNMP management server. The default *community_string* is *public*.

You can specify either an IPv4 address or an IPv6 address.

When you use the `Report> snmp show` command, *community_string* displays as follows:

```
public@mgmtserv1.veritas.com, public@mgmtserv2.veritas.com
```

For example, if using the IP address, enter the following:

```
Report> snmp add 10.10.10.10
```

```
Report> snmp add 2001:21::11
```

For example, if using the host name, enter the following:

```
Report> snmp add mgmtserv1.veritas.com
```

SNMP traps can be sent to multiple SNMP management servers.

To display the current list of SNMP management servers

- ◆ To display the current list of SNMP management servers, enter the following:

```
Report> snmp show
```

To delete an already configured SNMP management server from receiving SNMP traps

- ◆ To delete an already configured SNMP management server from receiving SNMP traps, enter the following:

```
Report> snmp delete snmp-mgmtserver-ipaddr
```

snmp-mgmtserver-ipaddr specifies the host name or the IP address of the SNMP management server.

To set the severity for SNMP traps to be sent

- ◆ To set the severity for SNMP traps to be sent, enter the following:

```
Report> snmp set severity value
```

where *value* indicates the severity for the SNMP trap to be sent.

See [“About severity levels and filters”](#) on page 199.

To set the filter level of SNMP traps

- ◆ To set the filter level for SNMP traps, enter the following:

```
Report> snmp set filter value
```

where *value* indicates the filter.

See [“About severity levels and filters”](#) on page 199.

To display the filter or the severity levels of SNMP traps to be sent

- ◆ To display the filter or the severity levels of SNMP traps to be sent, enter the following:

```
Report> snmp get filter|severity
```

To export the SNMP MIB file to a given URL

- ◆ To export the SNMP MIB file to a given URL, enter the following:

```
Report> snmp exportmib url
```

where *url* specifies the location the SNMP MIB file is exported to.

FTP and SCP URLs are supported.

If the *url* specifies a remote directory, the default file name is `access_mib.txt`.

Provisioning and managing Veritas Access file systems

- [Chapter 15. Creating and maintaining file systems](#)

Creating and maintaining file systems

This chapter includes the following topics:

- [About creating and maintaining file systems](#)
- [About scale-out file systems](#)
- [Erasure coding in a cluster file system \(CFS\) for NFS use case](#)
- [About encryption at rest](#)
- [Considerations for creating a file system](#)
- [Creating a file system](#)
- [Bringing the file system online or offline](#)
- [Listing all file systems and associated information](#)
- [Modifying a file system](#)
- [Managing a file system](#)
- [Destroying a file system](#)
- [Upgrading disk layout versions](#)

About creating and maintaining file systems

A Veritas Access environment consists of multiple nodes that can access and update files in the same Veritas file system at the same time. Many file systems can be supported at the same time. You create file systems on groups of disks called storage pools.

File systems consist of both metadata and file system data. Metadata contains information such as the last modification date, creation time, permissions, and so on. The total amount of the space that is required for the metadata depends on the number of files in the file system. A file system with many small files requires more space to store metadata. A file system with fewer larger files requires less space for handling the metadata.

When you create a file system, you need to set aside some space for handling the metadata. The space that is required is generally proportional to the size of the file system. For this reason, after you create the file system, a small portion of the space appears to be used. The space that is set aside to handle metadata may increase or decrease as needed. For example, a file system on a 1 GB volume takes approximately 35 MB (about 3%) initially to store metadata. In contrast, a file system of 10 MB requires approximately 3.3 MB (30%) initially for storing the metadata.

File systems can be increased or decreased in size. SmartTier functionality is also provided at the file system level.

About scale-out file systems

A scale-out file system consists of a set of on-premises file systems and set of cloud tier(s) all exposed in a single name space. One on-premises file system stores the metadata (including the attributes) and all the other file systems store the data. Data is distributed among the file systems using a consistent hashing algorithm. This separation of metadata and data allows the scale-out file system to scale linearly.

Veritas Access supports access to scale-out file systems using NFS-Ganesha, S3, CIFS and FTP.

Scale-out file system is the default file system for the LTR use case in Veritas Access 3340 Appliance.

Scale-out file system specifications:

- Twenty percent of a scale-out file system's size is devoted to the metadata file system.
- The maximum size of a metadata file system is 10 TB.
- The minimum size of a scale-out file system is 10 GB.
- The maximum size of a scale-out file system is 3 PB.
- To create a scale-out file system above 522 TB, you need to provide the file system size in multiples of 128 GB.
- You can grow a scale-out file system up to 3 PB.

- To create or grow a scale-out file system above 522 TB, you need to provide the file system size in multiples of 128 GB.

Note: Growing a scale-out file system beyond 522 TB creates additional data file systems (based on the grow size), and data movement is triggered from the old file systems to the newly added file systems, so that data is distributed evenly among all the data file systems.

- You can shrink the scale-out file system only if its size is less than 522 TB.
- Access the data present in a scale-out file system using NFS (both v3 and v4), S3 (supports both AWS signature version 2 and version 4), CIFS and FTP protocols.
 See [“Using the NFS-Ganesha server”](#) on page 109.
- Ability to tier infrequently accessed data to the cloud using the cloud as a tier feature:
 There can be only one on-premises tier.
 There can be up to eight cloud tiers per a scale-out file system.
 You can move data between cloud tiers, for example, moving data from Azure to Glacier.
 Configure policies to move data from or to on-premises or cloud tiers.
 Policies can be configured based on the access time, modification time, or pattern.
- Azure has a limitation of 500 TB per storage account. Azure users can have 200 storage accounts per subscription. A Scale-out file system supports adding multiple Azure storage accounts in a single tier. Effectively, you can attach 100 PB of Azure storage to a single tier. When multiple storage accounts are used, Veritas Access selects one of the storage accounts to store data in a round-robin manner.
- A scale-out file system can be configured for synchronous or asynchronous replication. Synchronous replication provides zero RPO for applications. Asynchronous replication provides non-zero RPO while providing improved performance compared to synchronous replication. This replication feature works even with cloud tier.

New data file systems are created when you grow the scale-out file system beyond 522 TB. The pool on which the scale-out file system is created is used to create these new file systems. There is also data movement to these new file systems so that data is distributed evenly among all the file systems (on-premises).

The following types of clouds can be added as storage tiers for a scale-out file system:

- Amazon S3
- Amazon Glacier
- Amazon GovCloud (US)
- Azure
- Google cloud
- Alibaba
- IBM Cloud Object Storage
- Veritas Access S3 and any S3-compatible storage provider

The data is always written to the on-premises storage tier and then data can be moved to the cloud using a tiering mechanism. File metadata including any attributes set on the file resides on-premises even though the file is moved to the cloud. This cloud as a tier feature is best used for moving infrequently accessed data to the cloud.

Amazon Glacier is an offline cloud tier, which means that data moved to Amazon Glacier cannot be accessed immediately. An EIO error is returned if you try to read, write, or truncate the files moved to the Amazon Glacier tier. If you want to read or modify the data, move the data to on-premises using tier move or using policies. The data is available after some time based on the Amazon Glacier retrieval option you selected.

When Amazon S3, AWS GovCloud(US), Azure, Google cloud, Alibaba, IBM Cloud Object Storage, Veritas Access S3 and any S3-compatible storage provider is used as the cloud tier, the data present on these clouds can be accessed any time (unlike in Amazon Glacier). An EIO error is returned if you try to write, or truncate the files moved to these clouds. If you want to modify the data, move the data to on-premises using tier move or using policies.

See the *Veritas Access Cloud Storage Tiering Solutions Guide* for more information.

Note: Veritas Access support the CIFS protocol with a scale-out file system.

See [“Configuring the cloud as a tier for scale-out file systems”](#) on page 253.

See [“Moving files between tiers in a scale-out file system”](#) on page 255.

Erasure coding in a cluster file system (CFS) for NFS use case

Erasure coding is configured with the *EC log* option for NFS use case. The CFS erasure-coded file system for NFS can be created using the Erasure Coded Storage Policy in the Veritas Access GUI and also through the Veritas Access CLISH.

See the *Veritas Access Online Help* for more information on the Erasure Coded Storage Policy.

CFS erasure-coded file system for NFS use case can be created through Veritas Access using the following command:

```
Storage> fs create ecoded fs_name size ncolumns nparity pool1
[, disk1,...] [blksize=<bytes>] [stripeunit=<kilobytes>] [encrypt={on|off}]
[stripe_aligned={yes|no}] [stripe_tag={node|disk}] [rotating_parity=
{yes|no}] [single_volume={yes|no}] [ecloglen={200M|500M|1G|2G|5G}]
[disk1 [, disk2,...]] [workload={virtualmachine|mediaserver}]
```

<i>stripe_unit</i>	Specifies the stripe width size to use for erasure-coded volume.
<i>encrypt</i>	Specifies whether the encrypted data is stored in erasure-coded volume
<i>stripe_aligned</i>	Specifies whether the object allocations are aligned to stripe length.
<i>stripe_tag</i>	Specifies tag for customize failure domain. It can be either disk or node.
<i>rotating_parity</i>	Specifies whether the striping is done with distributed parity.
<i>single_volume</i>	Specifies whether to create only a single volume in the erasure-coded file system. This parameter can be set only if <i>stripe_aligned</i> is set to no.
<i>ecloglen</i>	Specifies the per column log size for erasure-coded volume. This parameter can be set only if <i>stripe_aligned</i> is set to no.
<i>eclogdisk</i>	Specifies a comma-separated list of disks from all failure domains to be used for log allocation. This parameter can be set only if <i>stripe_aligned</i> is set to no.

workload

Specifies the type of workload using the file system.

For example:

```
Storage> fs create ecoded ecfs3 2g 3 1 pool1 blksize=1024 64k  
stripe_aligned=no stripe_tag=node rotating_parity=yes
```

See the *Veritas Access Command Reference Guide* for more details about creation of CFS erasure-coded file system for NFS.

You can also encrypt your file system using a Key Management Server (KMS).

See [“About encryption at rest”](#) on page 215.

About encryption at rest

Veritas Access provides advanced security for data at rest by the encryption of data volumes. Encryption is a technology that converts data or information into code that can be decrypted only by authorized users.

You can encrypt Veritas Access data volumes to:

- Protect sensitive data from unauthorized access.
- Retire disks from use or ship them for replacement without the overhead of secure wiping of content.

Encryption is implemented using the Advanced Encryption Standard (AES) cryptographic algorithm with 256-bit key size validated by the Federal Information Processing Standard (FIPS) Publication 140-2, (FIPS PUB 140-2) security standard.

When you create file systems in Veritas Access on encrypt volumes using this feature, Veritas Access generates a volume encryption key at the time of file system creation. This encryption key is encrypted (wrapped) using a different key that is retrieved from a Key Management Server (KMS). The wrapped key is stored with the volume record. The volume encryption key is not stored on disk.

Veritas Access supports the use of a KMS that conforms to the OASIS Key Management Interoperability Protocol (KMIP) version 1.1 specification.

During creation of encrypted volumes:

1. Veritas Access sends a key generation request to the configured KMS using the KMIP protocol.
2. KMS responds with a unique identifier. Veritas Access sends the identifier to KMS to obtain the key that is generated by KMS.

3. KMS responds with the key. Veritas Access generates the random volume encryption key, and encrypts it using the key that is provided by KMS.
4. Veritas Access stores the encrypted key and the KMS identifier in the volume record.

During startup of encrypted volumes:

1. Veritas Access retrieves the encrypted key and the KMS identifier from the volume record.
2. Veritas Access sends the identifier to KMS to obtain the key.
3. KMS responds with the key. Veritas Access decrypts the encrypted key (stored in the volume record) with the key provided by KMS.

Note: Veritas recommends that you use CPUs designed to support Advanced Encryption Standard Instruction Set (or the Intel Advanced Encryption Standard New Instructions (AES-NI) to improve performance.

Veritas recommends that you use IBM Secure Key Lifecycle Manager (SKLM), which supports KMIP protocol version 1.1, as a KMS server for this feature.

To register a Veritas Access cluster with the IBM SKLM KMS server.

- 1 Install the IBM SKLM server on any system in your environment. You can visit the [URL](#) to find the supported IBM SKLM servers with Veritas Access. Obtain KMS server's public certificate in `base64` format using its admin GUI console or the CLI.
- 2 In the Veritas Access GUI management console, go to **Settings> Services Management** to register the Veritas Access cluster with the KMS server.
- 3 Ensure that the time of the Veritas Access server and IBM SKLM server are in sync.
- 4 Select **Provide Key & Certificates** to generate self-sign certificates for the Veritas Access cluster. Provide the KMS server's public SSL certificate in the same window.
- 5 **Configure KMS Server** gets activated now. Select this tab to enter the KMS server-related details.
- 6 Use the IBM SKLM server's GUI-based management to accept the client request from the Veritas Access cluster and to accept its SSL keys.

You can use the `Storage> fs create` command to create the file system with `encrypt=on` option.


```
storage> fs create mirrored fs2 lg 2 pool1 protection=disk
      blksize=8192
      pdir_enable=no
      encrypt=on
```

You can use the storage encryption feature in the GUI by activating the secure data storage policy. You can add new NFS and CIFS shares using the activated policy.

Note: *encrypt=on* option can be used for all the file system layouts except *largefs*.

Considerations for creating a file system

The following sections describe the considerations and best practices for creating file systems.

Best practices for creating file systems

The following are the best practices for creating file systems:

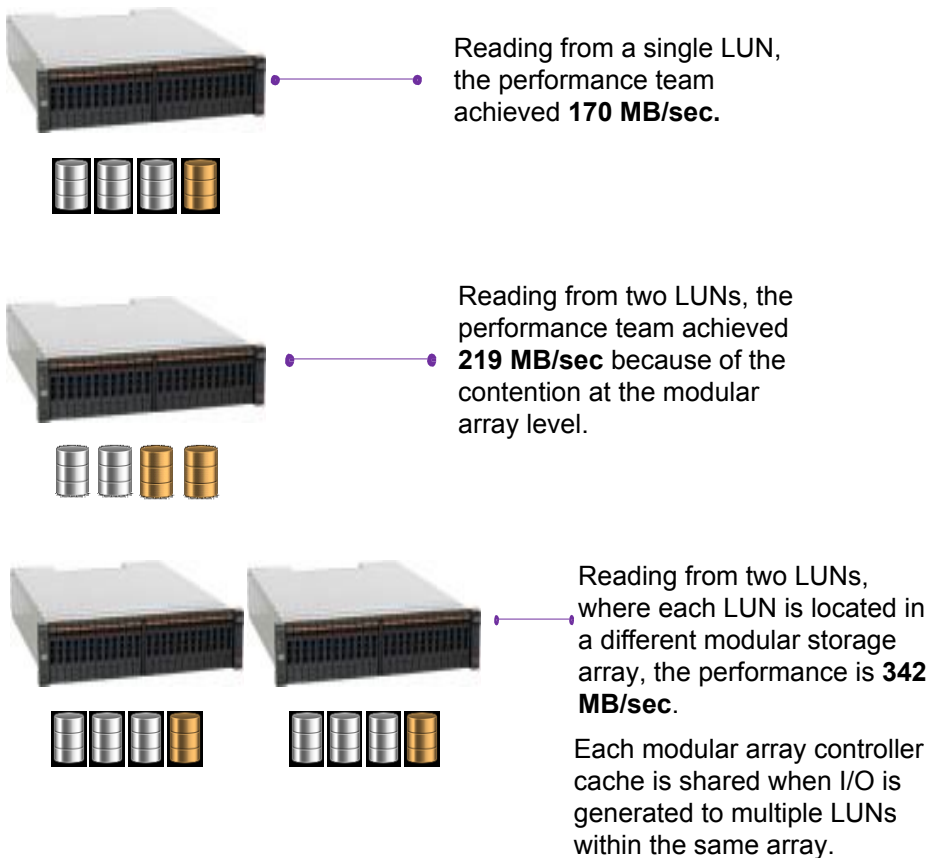
- Ensure all the disks (LUNs) in each storage pool have an identical hardware configuration.
Best performance results from a striped file system that spans similar disks. The more closely you match the disks by speed, capacity, and interface type, the better the performance you can expect. When striping across several disks of varying speeds, performance is no faster than that of the slowest disk.
- Create striped file systems rather than simple file systems when creating your file systems.
See [“About striping file systems”](#) on page 221.
- In a given storage pool, create all the file systems with the same number of columns.
- Ensure that the number of disks in each storage pool is an exact multiple of the number of columns used by the file systems created in that storage pool.
- Consider how many disks you need to add to your storage pool to grow your striped file systems.
A 5-TB file system using five columns cannot be grown in a storage pool containing 8*1-TB disks, despite having 3 TB of disk space available. Instead create the file system with either four or eight columns, or else add 2*1-TB disks to the pool. See further examples in the table.

Use case	Action	Result
----------	--------	--------

storage pool with eight disks of the same size (1 TB each)	Create a 5 TB striped file system with five columns.	You cannot grow the file system greater than 5 TB, even though there are three unused disks.
storage pool with eight disks of the same size (1 TB each)	Create a 5 TB striped file system with eight columns.	You can grow the file system to 8 TB.
storage pool with eight disks of the same size (1 TB each)	Create a 4 TB striped file system with four columns.	You can grow the file system to 8 TB.
storage pool with eight disks of the same size (1 TB each)	Create a 3 TB striped file system with three columns.	You cannot grow the file system to 8 TB.
storage pool with eight disks of the different sizes (3 are 500 GB each, and 5 are 2 TB each)	Create an 8 TB striped file system with eight columns.	You cannot create this 8-TB file system.

- Consider the I/O bandwidth requirement when determining how many columns you require in your striped file system.
Based on the disks you have chosen, I/O throughput is limited and potentially restricted. [Figure 15-1](#) describes the LUN throughput restrictions.
- Consider populating each storage pool with the same number of disks from each HBA. Alternatively, consider how much of the total I/O bandwidth that the disks in the storage pool can use.
If you have more than one card or bus to which you can connect disks, distribute the disks as evenly as possible among them. That is, each card or bus must have the same number of disks attached to it. You can achieve the best I/O performance when you use more than one card or bus and interleave the stripes across them.
- Use a stripe unit size larger than 64 KB. Performance tests show 512 KB as the optimal size for sequential I/O, which is the default value for the stripe unit. A greater stripe unit is unlikely to provide any additional benefit.
- Do not change the operating system default maximum I/O size of 512 KB.

Figure 15-1 LUN throughput - details on the LUN throughput restrictions



Choosing a file system layout type

Veritas Access allows you to create file systems with several layout types. The following table describes the layout types and their advantages.

Table 15-1 Types of volume layout

Layout type	Description
Simple	Arranges the disks sequentially and contiguously. A simple layout allows a file system to be created from multiple regions of one or more disks if there is not enough space on a single region of a disk.

Table 15-1 Types of volume layout (*continued*)

Layout type	Description
Striped	Spreads the data evenly across multiple disks. Stripes are equal-sized fragments that are allocated alternately and evenly to the disks. Throughput increases with the number of disks across which a file system is striped. Striping helps to balance I/O load in cases where high traffic areas exist on certain disks.
Mirrored	Mirrors the information contained in the file system to provide redundancy of data. For the redundancy to be useful, each mirror should contain disk space from different disks.
Mirrored-stripe	Configures a striped file system and then mirrors it. This requires at least two disks for striping and one or more other disks for mirroring (depending on whether the mirror is simple or striped). The advantages of this layout are increased performance by spreading data across multiple disks and redundancy of data.
Striped-mirror	Configures several mirrors as the columns of a striped file system. This layout offers the same benefits as a mirrored-stripe file system. In addition, it provides faster recovery as the failure of single disk does not force an entire striped mirror offline.

Determining the initial extent size for a file system

Veritas File System (VxFS) determines the size of the first extent that is allocated based on the first write to a new file. Normally, the first extent is the smallest power of 2 that is larger than the size of the first write. If that power of 2 is less than 8 KB (the default file system block size), the first extent that is allocated is 8 KB. After the initial extent is allocated, the file system increases the size of subsequent extents with each allocation as the file size is increased using extending writes.

The initial extent size is tunable, and can be changed using the `System> option modify tuneftab` command.

Increasing the initial extent size to a larger value helps to reduce file system fragmentation and improves I/O performance.

The best value for the initial extent size depends on the expected file sizes that are created by the application. The maximum value is 32768, which equates to a 256 MB extent allocation using the default 8 KB file system block size. Any over allocation of space is returned to the free space pool after the file is closed.

If the application creates a lot of small files with an exact size of 1 MB, then the initial extent size can be set to 128 (1 MB). If 1 MB is an approximate file size, then the initial extent size can be set to 64 (512 KB) instead. If most files are

approximately 1 GB or greater in size, then the maximum value of 32768 can be used.

About striping file systems

You can obtain huge performance benefits by striping (RAID-0) using software-defined storage (SDS). You achieve performance benefits regardless of the choice of LUN configuration in your storage hardware. Striping is useful if you need large amounts of data that is written to or read from physical disks, and consistent performance is important. SDS striping is a good practice for all Veritas Access use cases and workloads.

Veritas strongly recommends that you create striped file systems when creating your file system for the following reasons:

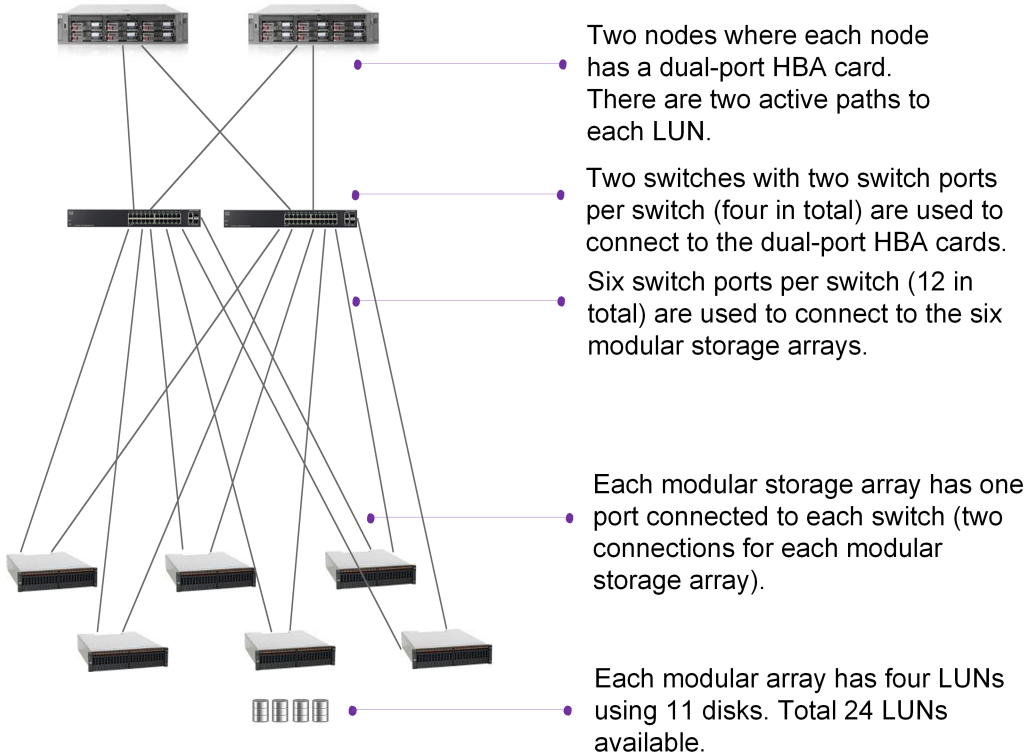
- Maximize the I/O performance.
- Proportion the I/O bandwidth available from the storage layer.
- Balance the I/O load evenly across multi-user applications running on multiple nodes in the cluster.

However there are also pitfalls to avoid.

The following information is essential before selecting the disks to include in your striped file system:

- Understanding of your hardware environment
- Storage capabilities and limitations (bottlenecks)
- Choice of LUNs (each LUN, or disk, equates to a column in a SDS-striped volume)

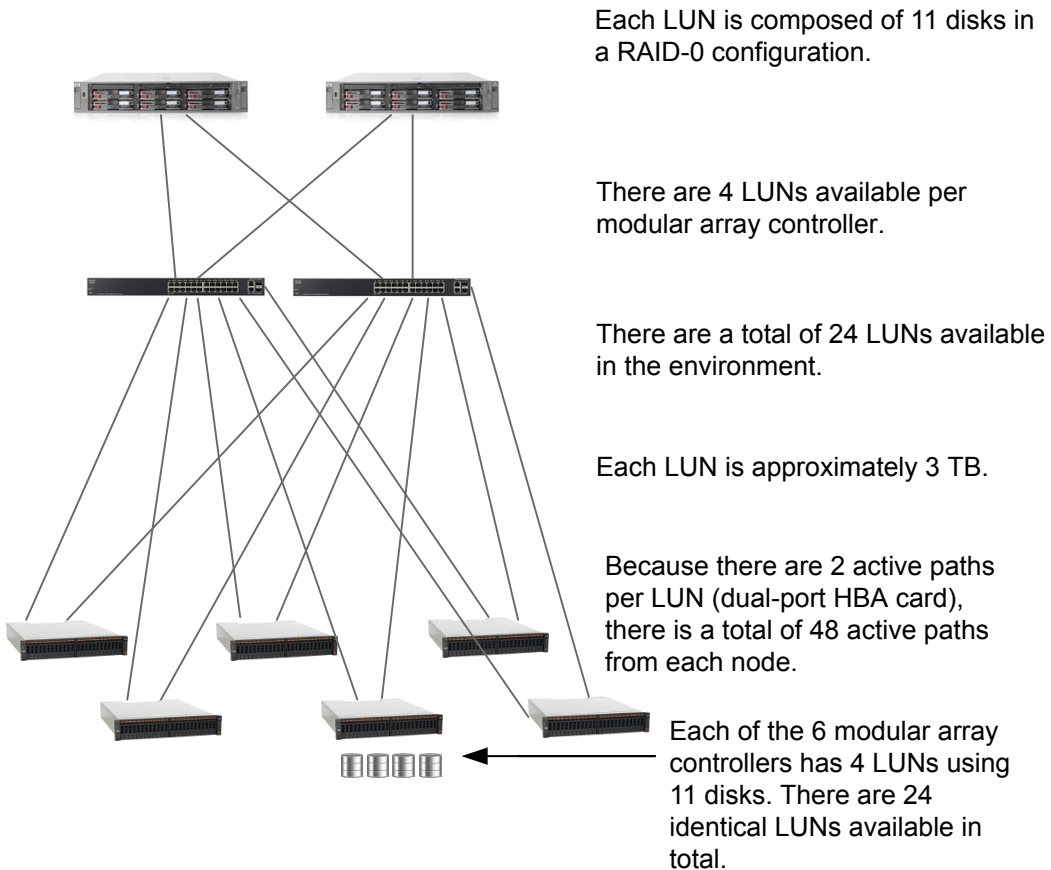
Figure 15-2 An example hardware configuration



An extreme example might be if one column (equal to one LUN) is composed of only hard disk drives (HDDs) in the storage array. All of the other columns in the same striped volume are composed of only SSDs in the storage array. The overall I/O performance bottlenecks on the single slower HDD LUN.

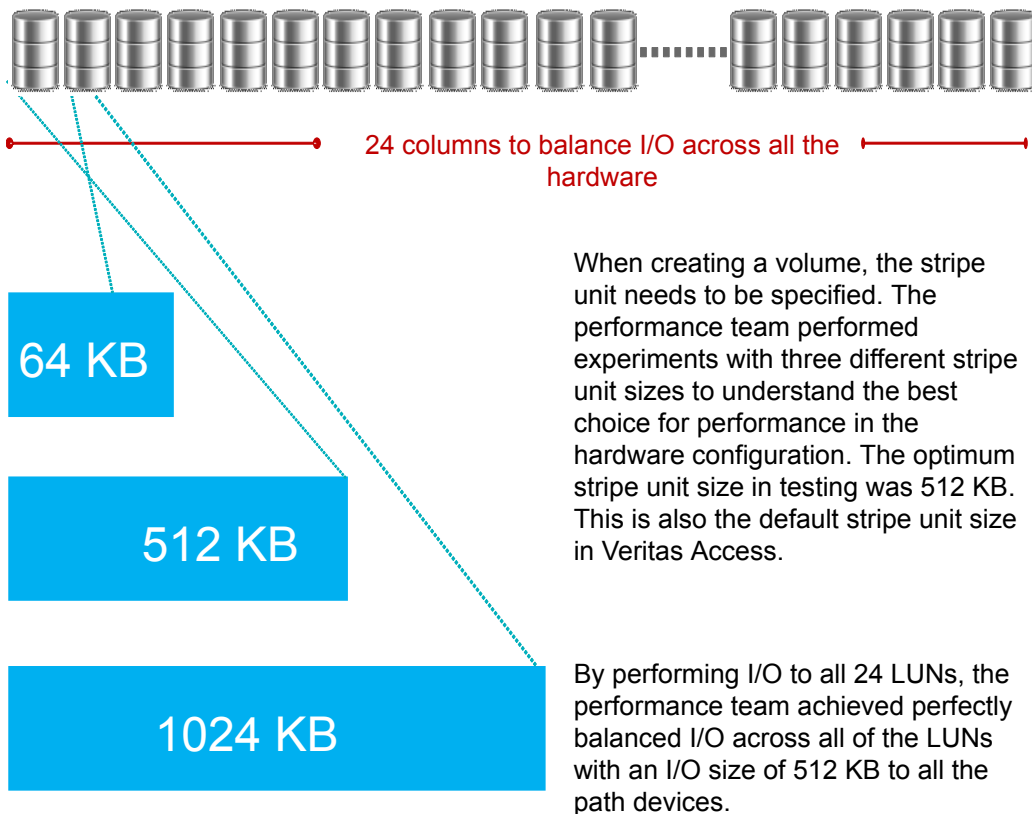
Understanding the LUN configuration and ensuring that all of the LUNs have an identical configuration is therefore essential for maximizing performance and achieving balanced I/O across all the LUNs.

Figure 15-3 LUN configuration



All 24 LUNs have an identical hardware configuration.

Figure 15-4 Volume configuration



The performance team created a volume with 24 columns to use the entire storage bandwidth in one file system. Veritas does not advise changing the operating system default maximum I/O size of 512 KB. The optimum stripe-unit size is 512 KB.

About creating a tuned file system for a specific workload

Veritas Access provides an easy way to create a well-tuned file system for a given type of workload.

You can use the newly created file system for the following common client applications:

- Virtual machine workloads
- Media server workloads

Streaming media represents a new wave of rich Internet content. Recent advancements in video creation, compression, caching, streaming, and other content delivery technology have brought audio and video together to the Internet as rich media. You can use Veritas Access to store your rich media, videos, movies, audio, music, and picture files.

See the `Storage> fs` man page for more information.

```
Storage> fs create pretuned media_fs 100g pool2 workload=mediaserver layout=striped 8
```

The `workload=mediaserver` option creates a file system called `media_fs` that is 100g in size in `pool2` striped across eight disks.

Note: You can select only one workload for a specified file system. You specify the workload when you create the file system, and you cannot change the workload after you have created the file system.

Virtual machine workloads

A virtual machine disk file, also known as a VMDK file, is a file held in the Veritas Access file system that represents a virtual disk for a virtual machine. A VMDK file is the same size as the virtual disk, so VMDK file sizes are typically very large. As writes are performed to the virtual disk within the virtual machine, the VMDK file is populated with extents in the Veritas Access file system. Because the VMDK files are large, they can become heavily fragmented, which gradually impedes performance when reading and writing to the virtual disk. If you create a file system specific to a virtual machine workload, Veritas Access internally tunes the file system to allocate a fixed extent size of 1MB for VMDK files. The 1MB block size significantly reduces both file system and VMDK file fragmentation while improving the virtual machine workload performance.

Media server workloads and tunable for setting write_throttle

Media server workloads involve heavy sequential reads and writes. Striping across multiple disks yields better I/O latency.

See [“Best practices for creating file systems”](#) on page 217.

For media server workloads, Veritas Access provides a tunable that can help restrict the amount of write I/O throughput. The tunable helps prevent the streaming of information (sequential reads) from being affected by other processes performing write I/O on the same NAS server. An example use case is as follows. You want to stream a movie, this is reading a file (sequential reads). You do not want the movie experience to pause due to buffering. Another user might be uploading new content to the same file system (the upload is writing data to a different file). The

uploading (writing) can cause the streaming (reading) to pause due to buffering. Veritas Access throttles the writing processes so that they do not consume too much of the system memory.

Each file system must tune the value `write_throttle` independently of other file systems. The default value is 0, which implies there is no `write_throttle`. The throttle is per file, so when writing to multiple files at the same time, the `write_throttle` threshold applies to each file independently.

Setting a non-zero value for a file system prevents the number of dirty memory pages that are associated with the file from increasing beyond the threshold. If you set a `write_throttle` value of 256, then writes to a file pause to flush the file to disk once 256 dirty memory pages have built up for the file. After the number of dirty pages for a file reaches the `write_throttle` threshold, further dirtying of pages is paused, and the file system starts flushing the file's pages to disk, even if free memory is available. Each memory page is 4KB of file data, so 256 pages is 1MB of file data. Setting a value for `write_throttle` means a writing thread pauses upon reaching the threshold (on the NAS server) while the file's dirty pages are flushed to disk, before resuming further writes to that file. Once flushed, the dirty pages become clean pages, which means the memory (the pages) can then be reused for perhaps pre-fetching data from disk for streaming reads. Setting a value for `write_throttle` helps prevent write I/O from consuming too much of the system memory.

Setting `write_throttle` requires some experimentation, which is why Veritas Access does not set a non-zero value by default. The valid range for `write_throttle` is 0 to 2048 pages. A good starting value for experimentation is 256.

About FastResync

The FastResync feature performs quick and efficient resynchronization of stale mirrors (a mirror that is not synchronized). FastResync optimizes mirror resynchronization by keeping track of updates to stored data that have been missed by a mirror.

When FastResync has been enabled, it does not alter how you administer mirrors. The only visible effect is that repair operations conclude more quickly.

About scale-out fsck

The scale-out fsck operation does the following:

- Checks the consistency of the metadata container, the data container, and the database, and repairs any inconsistencies.
- Checks if the metadata container and the data container are marked for full fsck. If yes, scale-out fsck performs a full fsck of the corresponding file systems.

Based on the actions taken by `fsck` on the individual file systems, the scale-out `fsck` operation repairs the inconsistencies in other parts of the scale-out file system.

See [“About scale-out file systems”](#) on page 211.

- Goes through all the file handles present in the database, and checks if the corresponding metadata container and the data container file handles are consistent with each other.

In some cases, full `fsck` might delete files from the data container. To maintain consistency, the corresponding files from the metadata container and the data container are removed, and the corresponding key is removed from the database.

```
Storage> fs fsck fs1
fsck of largefs fs1 is successful
```

Setting retention in files

The retention feature provides a way to ensure that the files are not deleted or modified until the retention period is applied on the files. You can set, clear, and show the retention period on files from the CLISH.

the file system should be created with `worm=yes` option to use the retention feature. See the `Storage> fs create` man page for more information.

To set retention:

```
Storage> fs retention set [path] [retention_period]
```

Where *path* is the specified file or directory on which retention is set. If the specified path is a directory, then retention is set on all the files that are currently present in the directory.

retention_period is the duration for which retention is set. It can be in either [1-9](d|D|m|M|y|Y) or mm-dd-yyyy format.

To show retention:

```
Storage> fs retention show [path]
```

Where *path* is the specified file on which retention is set.

To clear retention:

```
Storage> fs retention clear [path]
```

Where *path* is the specified file or directory on which retention is cleared. If the specified path is a directory, then it clears the retention on all the files that are currently present in the directory.

See the `Storage> fs` man page for detailed examples.

Setting WORM over NFS

When a file is committed as Write-Once-Read-Many (WORM), the data in the file can be read but cannot be altered. The retention time for a WORM file specifies the time period for which the file must be retained after it is committed to WORM storage. The file cannot be deleted till the retention period expires. Once the retention time period has expired, the storage system allows the deletion of the file.

The file system on the server should be created with `worm=yes` option as the per-file WORM feature is supported only on file systems created with this option.

See the `Storage> fs` create man page for more details.

Export the file system from the server and mount it on the client.

To enable WORM on a file over NFS

- 1 Change the access time of the file so that it has the same value as the period of retention.

```
# touch -at YYYYMMDDhhmm.ss <filename>
```

For example, if a file named `foo` has to be retained till 14th July, 2035 10:37:42pm, run the following command:

```
# touch -at 203507141037.42 foo
```

- 2 Mark the file as read-only by changing the permissions of the file.

For example, to make the file `foo` read-only, run the following command:

```
# chmod -w foo
```

On successful execution of the above two steps, WORM is enabled on the file, `foo` with *14th July, 2035 10:37:42pm* as the retention time.

About managing application I/O workloads using maximum IOPS settings

When multiple applications use a common storage subsystem, it is important to ensure that a particular application does not monopolize the storage bandwidth thereby impacting all the other applications using the same storage. It is also important to balance the application I/O requests in a way that allows all the applications to co-exist in a shared environment. You can address this need by

setting a maximum threshold on the I/O operations per second (MAXIOPS) for the file system.

The MAXIOPS limit determines the maximum number of I/Os processed per second collectively by the storage underlying the file system.

When an I/O request comes in from an application, it is serviced by the storage underlying the file system until the application I/O reaches the MAXIOPS limit. When the limit is exceeded for a specified time interval, further I/O requests on the application are queued. The queued I/Os are taken up on priority in the next time interval along with new I/O requests from the application.

You should consider the following factors when you set the MAXIOPS threshold:

- Storage capacity of the shared subsystem
- Number of active applications
- I/O requirements of the individual applications

Only application-based I/Os can be managed with MAXIOPS.

MAXIOPS addresses the use case environment of multiple applications using a common storage subsystem where an application is throttled because of insufficient storage bandwidth while another less critical application uses more storage bandwidth.

See the `maxiops` man pages for detailed examples.

Creating a file system

Veritas Access supports the following types of file systems:

- Cluster File System (CFS) - creates a standard CFS file system. The CFS file system is the default file system when creating a new file system.
- scale-out - creates a scale-out file system that manages a single namespace spanning over both on-premises storage as well as cloud storage.
See [“About scale-out file systems”](#) on page 211.

Veritas Access supports the following file system layouts:

- Simple - creates a file system of a specified size, and you can specify a block size for the file system.
- Mirrored - creates a mirrored file system with a specified number of mirrors along with a list of pools and online status.
- Mirrored-stripe - creates a mirrored-stripe file system with a specified number of columns, mirrors, pools, and protection options.

- Striped - creates a striped file system. A striped file system is a file system that stores its data across multiple disks rather than storing the data on just one disk.
- Striped-mirrored - creates a striped-mirror file system with a specified number of columns, mirrors, pools, and protection options.

The default block size is determined based on the size of the file system when the file system is created. For example, 1 KB is the default block size for up to a 2-TB file system size. Other default block sizes, 2 KB, 4 KB, and 8 KB are available for different ranges of file system sizes. If you create a 1-TB file system, and then increase it to 3 TB, the file system block size remains at 1 KB.

See the `storage_fs(1)` man page for detailed examples.

You can also create a file system for customized workloads.

See [“About creating a tuned file system for a specific workload”](#) on page 224.

Note: If the configuration file system creation fails, the originating file system request also fails. Veritas Access requires at least two disks for the mirrored configuration file system, and in case of Flexible Shared Storage (FSS), at least two nodes are required to be part of the storage pool.

To create a simple file system of a specified size

- ◆ Create a simple file system with a specified size.

```
Storage> fs create simple fs_name size
pool1[,disk1,...] [blksize=bytes]
```

To create a mirrored file system

- ◆ Create a mirrored file system.

```
Storage> fs create mirrored fs_name size nmirrors
pool1[,disk1,...] [protection=disk|pool] [blksize=bytes]
```

To create a mirrored-stripe file system

- ◆ Create a mirrored-stripe file system.

```
Storage> fs create mirrored-stripe fs_name size nmirrors
ncolumns pool1[,disk1,...]
[protection=disk|pool] [stripeunit=kilobytes] [blksize=bytes]
```

To create a striped-mirror file system

- ◆ Create a striped-mirror file system.

```
Storage> fs create striped-mirror fs_name size nmirrors
ncolumns pool1[,disk1,...]
[protection=disk|pool] [stripeunit=kilobytes] [blksize=bytes]
```

To create a striped file system

- ◆ Create a striped file system.

```
Storage> fs create striped fs_name size ncolumns
pool1[,disk1,...] [stripeunit=kilobytes]
[blksize=bytes]
```

To create a simple scale-out file system

- ◆ Create a simple scale-out file system.

```
Storage> fs create largefs simple fs_name size
pool1 [blksize=bytes]
```

You can create the same types of file system layouts for a scale-out file system as for a Cluster File System (CFS) file system.

fs_name	Specifies the name of the file system being created. The file system name should be a string. If you enter a file that already exists, you receive an error message and the file system is not created.
size	<p>Specifies the size of a file system.</p> <p>To create a file system, you need at least 10 MB of space.</p> <p>You can create a file system in the following units:</p> <ul style="list-style-type: none"> ■ MB ■ GB ■ TB <p>You can enter the units with either uppercase (10 M) or lowercase (10 m) letters.</p> <p>To see how much space is available on a pool, use the <code>Storage> pool free</code> command.</p> <p>See “About configuring storage pools” on page 68.</p>
nmirrors	Specifies the number of mirrors for the file system. You must enter a positive integer.

ncolumns	<p>Specifies the number of columns for the striped file system. The number of columns represents the number of disks to stripe the information across. If the number of columns exceeds the number of disks for the entered pools, an error message is displayed. This message indicates that there is not enough space to create the striped file system.</p>
pool1[,disk1,...]	<p>Specifies the pool(s) or disk(s) for the file system. If you specify a pool or disk that does not exist, you receive an error message. You can specify more than one pool or disk by separating the name with a comma. Do not include a space between the comma and the name.</p> <p>To find a list of pools and disks, use the <code>Storage> pool list</code> command. To find a list of disks, use the <code>Storage> disk list</code> command.</p> <p>The disk must be part of the pool or an error message is displayed.</p>
protection	<p>If you do not specify a protection option, the default is "disk."</p> <p>The available options for this field are:</p> <ul style="list-style-type: none"> ■ disk - Creates mirrors on separate disks. ■ pool - Creates mirrors in separate pools. If there is not enough space to create the mirrors, an error message is displayed, and the file system is not created. <p>The <code>protection=pool</code> option is not supported for an isolated pool.</p> <p>If the file system is created with the <code>protection=pool</code> option, then both the pools in question cannot be merged to create a single pool, which defeats the purpose of the <code>protection=pool</code> option.</p>
stripeunit=kilobytes	<p>Specifies a stripe unit (in kilobytes).</p> <p>The following are the possible values:</p> <ul style="list-style-type: none"> ■ 128 ■ 256 ■ 512 (default) ■ 1024 ■ 2048

`blksize=bytes`

Specifies the block size for the file system.

The following are the possible values in bytes:

- 1024
- 2048
- 4096
- 8192 (default)

Block sizes can affect the file size. For example, to create a file system greater than 32 TB, the block size needs to be 8192.

`pdir_enable`

Specifies if you want to enable a partition directory for the file system. By default, this feature is not enabled.

After a file system is created, the file system reserves some space for internal logging. Internal logging provides additional data integrity. Due to the space that is reserved for internal logging, the file system may appear to be used immediately after file system creation. The space that is reserved for internal logging increases with the number of nodes in the Veritas Access cluster.

Log file sizes for the file systems are as follows:

10 GB to 100 GB Log size = 60 MB per node

100 GB to 1 TB Log size = 100 MB per node

1 TB and above Log size = 256 MB per node

Bringing the file system online or offline

The `Storage> fs online` or `Storage> fs offline` command lets you mount (online) or unmount (offline) a file system. You cannot access an offline file system from a client.

To change the status of a file system

- ◆ To change the status of a file system, enter one of the following, depending on which status you use:

```
Storage> fs online fs_name
```

```
Storage> fs offline fs_name
```

where `fs_name` specifies the name of the file system that you want to mount (online) or unmount (offline). If you specify a file system that does not exist, an error message is displayed.

Listing all file systems and associated information

To list all file systems and associated information

- ◆ To list all file systems and associated information, enter the following:

```
Storage> fs list [fs_name]
```

where *fs_name* is optional.

If you do not enter a specified file system, a list of file systems is displayed.

Modifying a file system

You can modify a file system in the following ways:

See [“Adding or removing a mirror from a file system”](#) on page 234.

See [“Adding or removing a column from a file system”](#) on page 236.

See [“Increasing the size of a file system”](#) on page 237.

See [“Decreasing the size of a file system”](#) on page 239.

Adding or removing a mirror from a file system

A mirrored file system is one that has copies of itself on other disks or pools.

To add a mirror to a file system

- ◆ To add a mirror to a file system, enter the following:

```
Storage> fs addmirror fs_name pool1[,disk1,...]  
[protection=disk|pool]
```

<i>fs_name</i>	Specifies which file system to add the mirror. If the specified file system does not exist, an error message is displayed.
----------------	----------------------------------------------------------------------------------------------------------------------------

pool1[,disk1,...]	<p>Specifies the pool(s) or disk(s) to use for the file system. If the specified pool or disk does not exist, an error message is displayed, and the file system is not created. You can specify more than one pool or disk by separating the name with a comma, but do not include a space between the comma and the name.</p> <p>To find a list of existing pools and disks, use the <code>Storage> pool list</code> command.</p> <p>See “About configuring storage pools” on page 68.</p> <p>To find a list of the existing disks, use the <code>Storage> disk list</code> command.</p> <p>The disk needs to be part of the pool or an error message is displayed.</p>
protection	<p>The default value for the protection field is <code>disk</code>.</p> <p>Available options are:</p> <ul style="list-style-type: none"> ■ <code>disk</code> - if the protection is set to <code>disk</code>, then mirrors are created on separate disks. This flag only works for file systems of type mirrored, mirrored-striped, and striped-mirror. The disks may or may not be in the same pool. ■ <code>pool</code> - if the protection is set to <code>pool</code>, then mirrors are created in separate pools. This flag only works for file systems of type mirrored, mirrored-striped, and striped-mirror. If not enough space is available, then the file system creation operation fails.

To remove a mirror from a file system

- ◆ To remove a mirror from a file system, enter the following:

```
Storage> fs rmmirror fs_name [pool_or_disk_name]
```

fs_name	Specifies the file system from which to remove the mirror. If you specify a file system that does not exist, an error message is displayed.
pool_or_disk_name	Specifies the pool or the disk name to remove from the mirrored file system that spans the specified pools or disks. If a pool name is the same as the disk name, then the mirror present on the pool is deleted.

For a striped-mirror file system, if any of the disks are bad, the `Storage> fs rmmirror` command disables the mirrors on the disks that have failed. If no disks have failed, Veritas Access chooses a mirror to remove.

Adding or removing a column from a file system

You may want to add or remove a column from a file system in specific situations. Adding columns can help to perform more I/Os in parallel, so you may want to increase the number of columns in the file system.

Note: For a striped file system when you add a column, the layout that is displayed when you issue the `Storage> fs list` and `Storage> fs list fsname` commands may be different than the original layout of the file system while the relayout (addition of new columns) operation is in progress. The original file system layout is displayed when the layout operation is completed.

Note: Adding and removing a column to and from a file system involves a volume-level relayout. This is an I/O intensive operation. For a large file system, adding or removing columns takes a long time and can hurt application performance during this relayout period.

To add a specified number of columns to a file system

- ◆ To add a specified number of columns to a file system, enter the following:

```
Storage> fs addcolumn fs_name ncolumns pool_or_disk_name
```

fs_name	Specifies the file system for which you want to add additional columns.
ncolumns	<div>Specifies the number of columns that you want to add to the file system.</div> <div>Note: In the case of a striped file system, the number of the disks that are specified should be equal to the number of columns (<i>ncolumns</i>).</div> <div>Note: In the case of a mirrored-stripe and a striped-mirrored file system, the disks should be equal to (<i>ncolumns * number_of_mirrors_in_fs</i>).</div>
pool_or_disk_name	Specifies the pool or the disk name for the file system.

To remove a column from a file system

- ◆ To remove a column from a file system, enter the following:

```
Storage> fs rmcolumn fs_name
```

where *fs_name* is the name of the file system for which you want to remove the column.

Increasing the size of a file system

To increase (grow) the size of a file system, it must be online. If the file system is not online, an error message is displayed, and no action is taken.

Note: For a scale-out file system, you can increase the size of the primary tier only.

See [“About scale-out file systems”](#) on page 211.

To increase the size of a file system to a specified size

- ◆ To increase the size of a file system to a specified size, enter the following:

```
Storage> fs growto {primary|secondary} fs_name new_length  
[pool1[,disk1,...]]  
[protection=disk|pool]
```

If no pool is specified with the command, the disks for growing the file system can be taken from any available pool. The protection flag takes the default value of *disk* in this case. The value of the *protection* field cannot be set to *pool* when no pool is specified with the command. This operation may convert the layout of the file system if the command determines that the new file system is too large for the original layout.

To increase the size of a file system by a specified size

- ◆ To increase the size of a file system by a specified size, enter the following:

```
Storage> fs growby {primary|secondary} fs_name length_change  
[pool1[,disk1,...]]  
[protection=disk|pool]
```

If no pool is specified with the command, the disks for growing the file system can be taken from any available pool. The protection flag takes the default value of *disk* in this case. The value of the *protection* field cannot be set to *pool* when no pool is specified with the command. This operation may convert the layout of the file system if the command determines that the new file system is too large for the original layout.

`primary|secondary` Specifies the primary or the secondary tier.

`fs_name` Specifies the file system whose size is increased. If you specify a file system that does not exist, an error message is displayed.

`new_length` Expands the file system to a specified size. The size that you specify must be a positive number, and it must be bigger than the size of the existing file system. If the new file system is not larger than the size of the existing file system, an error message is displayed, and no action is taken.

This variable is used with the `Storage> fs growto` command.

`length_change` Expands the file system by a specified size. The size that you specify must be a positive number, and it must be lesser than the available space. If it exceeds the available space, an error message is displayed, and no action is taken.

This variable is used with the `Storage> fs growby` command.

`pool1[,disk1,...]` Specifies the pool(s) or disk(s) to use for the file system. If you specify a pool or disk that does not exist, an error message is displayed, and the file system is not resized. You can specify more than one pool or disk by separating the name with a comma; however, do not include a space between the comma and the name.

To find a list of existing pools and disks, use the `Storage> pool list` command.

See [“About configuring storage pools”](#) on page 68.

To find a list of the existing disks, use the `Storage> disk list` command.

The disk needs to be part of the pool or an error message is displayed.

`protection` The default value for the protection field is `disk`.

Available options are:

- `disk` - if the protection is set to `disk`, then mirrors are created on separate disks. This flag only works for file systems of type mirrored, mirrored-striped, and striped-mirror. The disks may or may not be in the same pool.
- `pool` - if the protection is set to `pool`, then mirrors are created in separate pools. This flag only works for file systems of type mirrored, mirrored-striped, and striped-mirror. If not enough space is available, then the file system creation operation fails.

Decreasing the size of a file system

You can decrease (shrink) the size of the file system.

Note: For a scale-out file system, you can decrease the size of the primary tier only. Also, you can decrease the size of the primary tier only if the size of the file system is less than 522 TB.

See [“About scale-out file systems”](#) on page 211.

To decrease the size of the file system, it must be online. If the file system is not online, an error message is displayed, and no action is taken.

You cannot decrease the size of a file system if a rollback exists. Delete the rollback first before using the `Storage> fs shrinkto` or `Storage> fs shrinkby` commands.

To decrease the size of a file system to a specified size

- ◆ To decrease the size of a file system, enter the following:

```
Storage> fs shrinkto {primary|secondary} fs_name new_length
```

To decrease the size of a file system by a specified size

- ◆ To decrease the size of a file system, enter the following:

```
Storage> fs shrinkby {primary|secondary} fs_name length_change
```

primary secondary	Specifies the primary or the secondary tier.
fs_name	Specifies the file system whose size decreases. If you specify a file system that does not exist, an error message is displayed.
new_length	Specifies the size to decrease the file system to. The size that you specify must be a positive number, and it must be smaller than the size of the existing file system. If the new file system size is not smaller than the size of the existing file system, an error message is displayed, and no action is taken.
length_change	Decreases the file system by a specified size. The size that you specify must be a positive number, and it must be smaller than the size of the existing file system. If the new file system size is not smaller than the size of the existing file system, an error message is displayed, and no action is taken.

Note: Decreasing the size of a file system can take a long time if there are many extents allocated in the shrink area, as these extents have to be relocated to other areas in the file system.

Managing a file system

The following sections detail about the managing a file system.

Defragmenting a file system

You can either defragment a file system now or you can schedule a defragment job for a file system.

Note: This is not supported for a scale-out file system.

To defragment a file system

- ◆ To defragment a file system, enter the following:

Storage> **fs defrag now *fs_name* *time* [*defrag_level*]**

fs_name Specifies the name of the file system that you want to defragment.

Note: The specified file system must be online before attempting to defragment the file system.

time Specifies the maximum time to run. The defragmentation options are processed until defragmentation is complete, or until the time limit expires. The time value should be larger than one minute.

Potential time value output and what the values mean:

- 10M - indicates 10 minutes
- 1H20M - indicates 1 hour and 20 minutes
- Infinite - indicates the defragmentation process continues to run until the defragmentation process is done completely.

There is no limit time.

defrag_level Specifies the defragmentation level such as `dir`, `extent`, or `all`.

To schedule a defragment job for a file system.

- 1 Create a defrag schedule job for a file system that reoccurs once a week:

```
Storage> fs defrag schedule create sched_name sched_duration \
minute [hour] [day_of_the_month] \
[month] [day_of_the_week]
```

<i>sched_name</i>	Specifies the the name of the schedule.
<i>sched_duration</i>	Specifies the duration of the defragmenatation job.
<i>minute</i>	Specifies the minute (0-59).
<i>hour</i>	Specifies the hour (0-23).
<i>day_of_the_month</i>	Specifies the day of the month (1-31).
<i>month</i>	Specifies the month of the year (1-12).
<i>day_of_the_week</i>	Specifies the day of the week (0-6 with 0=Sunday).

For example:

Create a defrag schedule called `schedule1` that runs at 11:00 pm every Saturday for a duration of 2 hours.

```
Storage> fs defrag schedule create schedule1 2 0 23 * * 6
```

The number 2 after `schedule1` is the duration of how long the defrag schedule will run. 0 indicates minutes and 23 is the hour for which the defrag schedule will run.

- 2 Show the defrag schedule details:

```
Storage> fs defrag schedule show sched_name
```

For example:

- 3 Start the defrag schedule job for a file system:

```
Storage> fs defrag schedule start fs_name sched_name
```

- 4 List the scheduled defrag job status for a file system:

```
Storage> fs defrag schedule list fs_name
```

Checking and repairing a file system

The `Storage> fs fsck` command lets you check and repair a file system while the file system is offline.

The `Storage> fs fsck` command tries to perform a normal fsck (check and repair) of the file system first, but if the `fullfsck` option is set, the command proceeds depending on the input that is provided by the user.

In most cases, a normal fsck (only log replay) is sufficient to repair a file system. In cases where there is structural damage to the file system's metadata, a full fsck of the file system may be necessary to repair the file system.

Warning: Using the `Storage> fs fsck` command on an online file system can damage the data on the file system. Only use the `Storage> fs fsck` command on a file system that is offline.

Note: When running the `Storage> fs fsck` command, you may encounter a process of `Unknown`. The `Unknown` process is normal, since there is no process printed as output when running a normal fsck using `Storage> fsck fs_name`. Full fsck is run only if the normal fsck fails. In the support mode, if you are running a full fsck, Veritas Access records that status in an internal database/file.

To check and repair a file system

- ◆ To check and repair a file system, enter the following:

```
Storage> fs fsck fs_name
```

where `fs_name` specifies the file system for which you want to check and repair.

Configuring FastResync for a file system

If the power fails or a switch fails, mirrors in a file system may not be in a consistent state.

The `Storage> fs setfastresync(FastResync)` command performs quick and efficient resynchronization of stale mirrors.

Note: You must have at least two mirrors on the file system to enable FastResync.

To enable the FastResync option

- ◆ To enable FastResync for a file system, enter the following:

```
Storage> fs setfastresync fs_name [pool_or_disk_name]
```

fs_name Specifies the name of the file system for which to enable FastResync.

pool_or_disk_name Specifies the pool or the disk name to resynchronize from the mirrored file system that spans the specified pool or disk.

You can also enable FastResync for a tier of a file system.

```
Storage> tier setfastresync fs_name [pool_or_disk_name]
```

fs_name Specifies the name of the file system for which to enable FastResync.

pool_or_disk_name Specifies the pool or the disk name to resynchronize from the mirrored file system that spans the specified pool or disk.

Disabling the FastResync option for a file system

You can disable the FastResync option for a file system.

Note: When instant rollbacks exist for a volume, you cannot disable the FastResync option for a file system.

To disable the FastResync option

- ◆ To disable the FastResync option for a file system, enter the following:

```
Storage> fs unsetfastresync fs_name
```

where:

<i>fs_name</i>	Specifies the name of the file system for which to disable FastResync. If you specify a file system that does not exist, an error message is displayed.
----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------

You can also disable FastResync for a tier of a file system.

```
Storage> tier unsetfastresync fs_name
```

where:

<i>fs_name</i>	Specifies the name of the file system for which to disable FastResync. If you specify a file system that does not exist, an error message is displayed.
----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------

Checking and resynchronizing stale mirrors

You can check if there is a stale mirror on any of your file systems. If there is a stale mirror, the stale mirror needs to be resynchronized, and the resynchronization process needs to be verified.

To check if there are stale mirrors on your file systems

- ◆ To check if your file systems contain a stale mirror, enter the following:

```
Storage> fs checkmirror
```

To resynchronize all stale mirrors or a stale mirror for a specified file system

- ◆ To resynchronize all stale mirrors or a stale mirror for a specified file system, enter the following:

```
Storage> fs resync [fs_name]
```

where *fs_name* is the name of the specified file system where you want to resynchronize for stale mirrors.

If you do not include *fs_name*, you resynchronize all the stale mirrors for all your file systems.

To verify the resynchronization process for your stale mirrors

- ◆ To verify the resynchronization process for your stale mirrors, enter the following:

```
Storage> fs checkresync
```

Note: If a column addition to a file system is in progress, the output of the `Storage> fs checkresync` command will include RELAYOUT status.

Setting file system alerts

For a file system to run efficiently, you should always reserve some space for the file system rather than using 100% of the space. You can set file system alerts based on file system or snapshot usage. You can set the alert based on the number of inodes used, file system space used, or snapshot usage.

File system alerts can be displayed by using the `Report> showevents` command.

To set file system alerts

- ◆ To set file system alerts, enter the following:

```
Storage> fs alert set numinodes | numspace | fullspace | fullinodes
value [fs_name,...] [snapshot_name]
```

numinodes	When setting the alert for <code>numinodes</code> , <i>value</i> is the number of inodes used. The default alert value for <code>numinodes</code> is set at 0. An alert will not be sent until you set it to a different value.
numspace	When setting the alert for <code>numspace</code> , <i>value</i> is the percentage you want to set to trigger the alert. By default, the alert is sent at 80%. If you do not specify a file system name, the default value is modified.
fullspace	<code>fullspace</code> is the tunable for setting an alert if the file system becomes full. When file system usage is above the limit set by the <code>fullspace</code> tunable, all the NFS/CIFS shares on the file system are automatically changed to read-only to prevent the file system from becoming full again. When you grow the file system or delete some files to free up space, the NFS/CIFS shares are automatically changed back to read-write (there might be a delay of up to five minutes) for the change to occur.

Note: The file system size is checked every five minutes. During this five-minute interval, if the usage of the file system grows to more than 80%, the NFS/CIFS shares are changed to read-only. If the file system is small and write I/O is fast, then the file system can be filled up to 100% before being changed to read-only. This is by design.

By default, the `fullspace` tunable is set to 0, which means that the `fullspace` tunable is disabled.

fullinodes	<code>fullinodes</code> is the tunable for setting an alert if the file system becomes full. When inodes on a file system reach the limit of <code>fullinodes</code> , the NFS/CIFS shares on the file system are automatically changed to read-only. After the file system is changed to read-only, you need to delete some files from the file system, and a remount of the file system may be required for the NFS/CIFS shares to be changed to read-write. By default, the <code>fullinodes</code> tunable is set to 0, which indicates that the <code>fullinodes</code> tunable is disabled.
fs_name	Name of the file system for which you want to set the file system alerts. <i>fs_name</i> is optional. To specify multiple file systems, use commas to separate the file system names.
snapshot_name	Name of the snapshot for which you want to set the file system alert. Note: The following are reserved words for <code>snapshot_name</code> : <code>flags</code> , <code>ctime</code> , and <code>mtime</code> .

Displaying file system alert values

You can display the current disk space usage and the set alert value. A `D` beside the value indicates that the value is the default value used throughout the system.

To display file system alert values

- ◆ To display file system alert values, enter the following:

```
Storage> fs alert show
```

Removing file system alerts

You can remove the alerts set on a file system. If you remove an alert on any file system, you receive alerts for the file systems based on the default values.

To remove file system alerts

- ◆ To remove file system alerts, enter the following:

```
Storage> fs alert unset numinodes | numspace | fullspace | fullinodes
[fs_name,...] [snapshot_name]
```

fs_name Name of the file system for which you want to remove the file system alert. *fs_name* is optional.

When the fullspace/fullinodes tunables are unset (set to 0), the shares that were changed to read-only due to file system high usage are changed back to read-write mode immediately.

snapshot_name Name of the snapshot for which you want to remove the file system alert.

Note: The following are reserved words for *snapshot_name*: *flags*, *ctime*, and *mtime*.

See “[Setting file system alerts](#)” on page 246.

Destroying a file system

The `Storage> fs destroy` command unmounts a file system and releases its storage back to the storage pool. You cannot destroy the file systems that CIFS or NFS share.

To destroy a file system

- ◆ To destroy a file system, enter the following:

```
Storage> fs destroy fs_name
```

where *fs_name* specifies the name of the file system that you want to destroy.

Upgrading disk layout versions

In this release, you can create and mount only file systems with disk layout Version 8 and above.

To upgrade the disk layout versions

- 1 Ensure the file system is online before proceeding.

You can find information about the file system version, enter the following:

```
Storage> fs list file_system_name
```

- 2 To upgrade a file system to the current layout, enter the following:

```
Storage> fs upgrade file_system_name
```

Configuring cloud storage

- [Chapter 16. Configuring the cloud gateway](#)
- [Chapter 17. Configuring cloud as a tier](#)

Configuring the cloud gateway

This chapter includes the following topics:

- [About the cloud gateway](#)
- [Configuring the cloud gateway](#)

About the cloud gateway

You can configure Veritas Access as a gateway to cloud storage. You can register cloud subscriptions to your Veritas Access cluster. Multiple cloud subscriptions can be attached, so you need to assign a service name to each subscription. You can then use the service name to attach the subscription to a scale-out file system as a storage tier.

The following clouds can be added as storage tiers for a scale-out file system:

- Amazon S3
- Alibaba
- AWS
- AWS Gov Cloud (US)
- Azure
- Google
- IBM Cloud Object Storage
- Veritas Access S3
- Any S3-compatible storage provider

The cloud as a tier feature lets you have hybrid storage that uses both on-premises storage and public or private cloud storage. After the gateway and tier are configured, you can use the cloud as a tier feature to move data between the cloud and the on-premises storage. The files in the cloud, like the files on the on-premises storage, are accessible using the NFS, S3, CIFS and FTP protocols. Access of the data present in the cloud tier is transparent to the application.

See [“Configuring the cloud as a tier for scale-out file systems”](#) on page 253.

Before you provision cloud storage, you set up cloud subscriptions. To set up the cloud gateway, you attach cloud subscriptions to your Veritas Access cluster. You need to have the subscription credentials to add the cloud subscriptions. You need different subscription credentials based on your cloud provider.

Configuring the cloud gateway

A cloud gateway enables you to register one or more cloud services to the cluster. You can add the cloud service as a cloud tier for a scale-out file system. Whenever you are adding a tier to a scale-out file system, you need to add a cloud service for the cloud tier. You require a cloud service for a scale-out file system.

To configure the cloud service for scale-out file systems

- 1 Add the cloud service.

```
Storage> cloud addservice service_name service_provider=AWS
```

You are prompted to provide Amazon S3 subscription credentials.

- 2 Display the added cloud services.

```
Storage> cloud listservice service_name
```

- 3 Remove the cloud service.

```
Storage> cloud removeservice service_name
```

If any scale-out file system has a cloud tier associated with the service, the remove cloud service operation fails. Remove all the tiers from all the scale-out file systems before removing the cloud service.

Configuring cloud as a tier

This chapter includes the following topics:

- [Configuring the cloud as a tier for scale-out file systems](#)
- [Moving files between tiers in a scale-out file system](#)
- [About policies for scale-out file systems](#)
- [Obtaining statistics on data usage in the cloud tier in scale-out file systems](#)

Configuring the cloud as a tier for scale-out file systems

You can configure the following clouds as a storage tier with a scale-out file system:

- Amazon S3
- Amazon Glacier
- AWS GovCloud(US)
- Azure
- Alibaba
- Google cloud
- IBM Cloud Object Storage
- Veritas Access S3
- Any S3-compatible storage provider

The data is always written to the on-premises storage tier and then data can be moved to the cloud tier by setting up policies. The policies can be configured based on access time, modification time, and pattern of the files. File metadata including any attributes set on the file resides on-premises even though the file is moved to

the cloud. The cloud as a tier feature is best used for moving infrequently accessed data to the cloud.

See [“About scale-out file systems”](#) on page 211.

Warning: When any cloud service is used as a cloud tier for a file system, Veritas Access exclusively owns all the buckets and the objects created by Veritas Access. Any attempt to tamper with these buckets or objects outside of Veritas Access corrupts the files represented by those modified objects.

See [“Moving files between tiers in a scale-out file system”](#) on page 255.

See the `storage_cloud(1)` man page for detailed examples.

See the `storage_tier(1)` man page for detailed examples.

Warning: Veritas Access cannot add a cloud tier if the clock on the Veritas Access system is more than 15 minutes out-of-sync with the actual time. To ensure that the Veritas Access clock time is accurate, configure an NTP server or use the `System> clock set` command.

To configure the cloud as a tier for scale-out file systems

- 1 Display the available cloud services.

```
Storage> cloud listservice service_name
```

If the cloud service is not listed, you may need to add the cloud subscription to the cluster.

See [“About the cloud gateway”](#) on page 251.

- 2 Add the cloud as a tier to a scale-out file system.

```
Storage> tier add cloud fs_name tier_name service_name
region S3 | Glacier
```

Amazon AWS has standard regions defined. Based on the region you choose in Veritas Access, AWS storage is served through that region. To get better performance, always select the closest geographic region.

- 3 Verify that the cloud tier is configured on the specified scale-out file system.

```
Storage> tier list fs_name
```

To remove the cloud tier

- ◆ Remove the cloud tier.

```
Storage> tier remove fs_name tier_name
```

If there are any files present in the cloud tier, the remove cloud tier operation fails. Move the files back to the primary tier before removing the cloud tier.

Moving files between tiers in a scale-out file system

By default, a scale-out file system has a single tier (also known as a primary tier), which is the on-premises storage for the scale-out file system. You can add a cloud service as an additional tier. After a cloud tier is configured, you can move data between the tiers of the scale-out file system as needed. There can be up to eight cloud tiers per a scale-out file system. For example, you can configure Azure and AWS Glacier as two tiers and move data between these clouds.

Use the commands in this section to move data as a one-time operation. For example, if you have just set up a cloud tier, and you want to move some older data to that tier.

If you want to specify repeatable rules for maintaining data on the tiers, you can set up a policy for the file system.

See [“Creating and scheduling a policy for a scale-out file system”](#) on page 261.

You can specify the following criteria to indicate which files or directories to move between tiers:

- file or directory name pattern to match
- last accessed time (`atime`)
- last modified time (`mtime`)

Because a scale-out file system can be large, and the size of the files to be moved can be large as well, the `Storage> tier move` command lets you perform a dry run.

See the `storage_tier(1)` man page.

To move data between storage tiers in a scale-out file system

- 1 (Optional) Perform a dry run to see which files would be moved and some statistics about the move.

```
Storage> tier move dryrun fs_name source_tier destination_tier pattern
[atime condition] [mtime condition]
```

The dry run starts in the background. The command output shows the job ID.

- 2 Move the files that match *pattern* from *source_tier* to *destination_tier* based on the last accessed time (*atime*) or the last modified time (*mtime*).

```
Storage> tier move start fs_name source_tier destination_tier pattern
[atime condition] [mtime condition]
```

pattern is required. To include all the files, specify * for *pattern*.

The *condition* for *atime* or *mtime* includes an operator, a value, and a unit. Possible operators are the following: <, <=, >, >=. Possible units are m, h, and d, indicating minutes, hours, and days.

The name of the default tier is *primary*. The name of the cloud tier is specified when you add the tier to the file system.

The move job starts in the background. The command output shows the job ID.

Examples:

Move the files that match *pattern* and that have not been accessed within the past 100 days to the cloud tier.

```
Storage> tier move start lfs1 primary cloudtier1 pattern
atime >100d
```

Move the files that match *pattern* and that have been accessed recently in the past 30 days to the specified tier.

```
Storage> tier move start lfs1 cloud_tier primary pattern atime <=30d
```

Move the files that match *pattern* and that have not been modified within the past 100 days to the cloud tier.

```
Storage> tier move start lfs1 primary cloud_tier pattern
mtime >=100d
```

Move only the files that match *pattern* and that were modified within the last three days from the cloud tier to the primary tier.

```
Storage> tier move start lfs2 cloud_tier primary pattern mtime >=3d
```

Move all files to the primary tier.

```
Storage> tier move start lfs2 cloud_tier primary *
```

- 3 View the move jobs that are in progress in the background. This command lists the job IDs and the status of the job.

```
Storage> tier move list
```

Job	Fs name	Source Tier	Destination Tier	Pattern	Atime	Mtime	State
1473684478	largefs1	cloudtier	primary	/vx/largefs1/*	>120s	-	not running
1473684602	largefs1	cloudtier	primary	/vx/largefs1/*	-	-	scanning

- 4 View the detailed status of the data movement for the specified job ID.

```
Storage> tier move status jobid
```

For example:

```
Storage> tier move status 1473682883
```

```
Job run type:      normal
Job Status:        running
Total Data (Files): 4.0 G (100)
Moved Data (Files): 100.0 M (10)
Last file visited: /vx/fstfs/10.txt
```

- 5 If required, you can abort a move job.

```
Storage> tier move abort jobid
```

About policies for scale-out file systems

When a scale-out file system includes a cloud tier, you can use policies to control the movement of data between the on-premises storage and the cloud tier. A policy is a set of rules defined for a file system for deleting data or moving data. If you want to specify repeatable rules for maintaining data on the tiers, you can set up a policy for the file system.

Each rule defines the following criteria:

- what action should be taken (move or delete)
- when the data should be moved or deleted based on the access time or modified time of the file
- which data should be moved based on the pattern matching for the files and directories.

See [“About pattern matching for data movement policies”](#) on page 259.

Each file system can have more than one rule, though you should be careful not to create rules that conflict or cause looping.

To move or delete the data, you run the policy. When you run a policy, the rules are applied to the data at the time the policy is run. The policy does not run automatically. You can attach a schedule to a policy to have it run automatically at the specified times.

See [“Creating and scheduling a policy for a scale-out file system”](#) on page 261.

About pattern matching for data movement policies

Within a policy, you can use a pattern to specify that the rule applies to file names or directory names that match the pattern. By using a pattern, you do not need to know the exact file name in advance; the files that match the pattern are selected dynamically.

A pattern uses special characters, which are case sensitive. There are the following types of patterns:

- Directory patterns
A pattern that ends with a slash (/) is matched only against directories.
- File patterns
A pattern that does not end with a slash (/) is matched only against files.

The following is a list of supported special characters and their meanings:

- | | |
|-------------------|--------------------------------------------|
| * (asterisk) | Matches any character any number of times. |
| ? (question mark) | Matches any single character. |

** (two asterisks)	<p>Matches across child directories recursively.</p> <p>The pattern <code>fs1/**/*.pdf</code> will match <code>.pdf</code> file names present after first sub-directory of <code>fs1/</code>. For example, if the following files exist:</p> <pre>fs1/dir1/a.pdf fs1/dir2/b.pdf fs1/dir3/dir4/c.pdf dir5/d.pdf e.pdf</pre> <p>then the pattern <code>fs1/**/*.pdf</code> will match only <code>a.pdf</code> and <code>b.pdf</code>. It will not match <code>c.pdf</code>, <code>d.pdf</code> and <code>e.pdf</code>.</p> <p>The pattern <code>fs1/**/*.pdf</code> will match <code>.pdf</code> files in any directory after <code>fs1</code>. For the above file list, it will match all of the files: <code>a.pdf</code>, <code>b.pdf</code>, and <code>c.pdf</code>.</p> <p>The pattern <code>**/*.pdf</code> will match all the <code>.pdf</code> files in the whole file system. For the above file list, it will match all of the files: <code>a.pdf</code>, <code>b.pdf</code>, <code>c.pdf</code>, <code>d.pdf</code> and <code>e.pdf</code>.</p>
[] (square brackets)	<p>Matches either range or set of characters. <code>[0-5]</code> will match any character in range of 0 to 5. <code>[a-g]</code> will match any character in range of a to g. <code>[abxyz]</code> will match any one character out of a,b,x,y,z.</p>
! (exclamation point)	<p>Can be used as the first character in a range to invert the meaning of the match. <code>![0-5]</code> will match any character which is not in range of 0 to 5.</p>
\ (backslash)	<p>Can be used as an escape character. Use this to match for one of the above pattern matching characters to avoid the special meaning of the character.</p>

About schedules for running policies

A schedule is specified in a format similar to the UNIX `crontab` format. The format uses five fields to specify when the schedule runs:

minute	<p>Enter a numeric value between 0-59, or an asterisk (*), which represents every minute. You can also enter a step value (*/*), or a range of numbers separated by a hyphen.</p>
--------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

hour	Enter a numeric value between 0-23, or an asterisk (*), which represents every hour. You can also enter a step value (*/x), or a range of numbers separated by a hyphen.
day_of_the_month	Enter a numeric value between 1-31, or an asterisk (*), which represents every day of the month. You can also enter a step value (*/x), or a range of numbers separated by a hyphen.
month	Enter a numeric value between 1-12, or an asterisk (*), which represents every month. You can also use the names of the month. Enter the first three letters of the month (you must use lower case letters). You can also enter a step value (*/x), or a range.
day_of_the_week	Enter a numeric value between 0-6, where 0 represents Sunday, or an asterisk (*), which represents every day of the week. You can also enter the first three letters of the week (you must use lower case letters). You can also enter a step value (*/x), or a range.

A step value (*/x) specifies that the schedule runs at an interval of x. The interval should be an even multiple of the field's range. For example, you could specify */4 for the hour field to specify every four hours, since 24 is evenly divisible by 4. However, if you specify */15, you may get undesired results, since 24 is not evenly divisible by 15. The schedule would run after 15 hours, then 7 hours.

A range of numbers (two values separated by a hyphen) represents a time period during which you want the schedule to run.

Examples: To run the schedule every two hours every day:

```
0 */2 * * *
```

To run the schedule on 2:00 a.m. every Monday:

```
* 2 * * * 1
```

To run the schedule at 11:15 p.m. every Saturday:

```
15 23 * * * 6
```

Creating and scheduling a policy for a scale-out file system

By default, a scale-out file system has a single disk tier, which is the on-premises storage for the scale-out file system. You can add a cloud service as an additional tier. After a cloud tier is configured, you can move data between the tiers of the scale-out file system as needed.

Use policies to define a set of data movement rules for the scale-out file system. Each file system can include a policy for deletion and a policy for data movement between tiers.

Be careful when specifying the criteria for moving files. Conflicting policies may cause data to move from one tier to another tier. A best practice is to use policies with a smaller data set first before applying those policies to file systems using a schedule.

A data movement policy can use the following criteria to indicate which files or directories to move between tiers:

- pattern
- atime
- mtime

You can also perform a dry run of a policy.

See the `storage_fs(1)` policy section of the manual page for detailed examples.

To create a policy

- 1 Create a data movement policy `policy1` for file system `fs1` to move the files with file name extensions of `.txt` and `.pdf` from the primary tier (disk tier) to `tier1` (cloud tier), which did not get accessed or modified for the last two days.

```
Storage> fs policy add operation=move policy1 fs1 primary tier1 *.txt,*  
.pdf atime >2d mtime >2d  
ACCESS policy SUCCESS V-288-0 Policy policy1 for fs fs1 added  
successfully.
```

- 2 Retrieve data from Amazon Glacier. Create a policy `pol1` to move all the files with the file name extension of `.txt` from Amazon Glacier to the primary tier using the Bulk retrieval option.

Files are copied to on-premises and then deleted from Amazon Glacier. The time when the files are available on-premises depends on the type of retrieval option selected.

```
Storage> fs policy add operation=move pol1 gfs2 gtier primary  
retrieval_option=Bulk \*.txt
```

- 3 Create a data deletion policy policy2 for file system fs1 to move the files with file name extensions of .txt and .pdf from tier1 (cloud tier), which did not get accessed or modified for the last two days.

```
Storage> fs policy add operation=delete policy2 fs1 tier1 \*.txt,
\*.pdf atime >2d mtime >2d
ACCESS policy SUCCESS V-288-0 Policy policy2 for fs fs1 added
successfully.
```

- 4 Modify data movement policy policy1 for file system fs1 to move the files with the file name extension of .doc, which did not get accessed or modified for the last three days.

```
Storage> fs policy modify policy1 \*.doc atime >3d mtime >3d
ACCESS policy SUCCESS V-288-0 Policy policy1 modified
successfully.
```

- 5 List all the policies.

```
Storage> fs policy list
```

Name	FS name	Action	Source Tier	Destination Tier	Retrieval Option	Pattern
policy2	fs1	delete	tier1	-	Standard	*.txt, *.pdf
policy1	fs1	move	primary	tier1	Standard	*.doc

Atime	Mtime	State
>2d	>2d	not running
>3d	>3d	running

- 6 List all the policies set for file system fs1.

```
Storage> fs policy list fs1
```

Name	FS name	Action	Source Tier	Destination Tier	Retrieval Option	Pattern
policy2	fs1	delete	tier1	-	Standard	*.txt, *.pdf
policy1	fs1	move	primary	tier1	Standard	*.doc

Atime	Mtime	State
>2d	>2d	running
>3d	>3d	not running

7 Delete policy policy1 set for file system fs1.

```
Storage> fs policy delete policy1 fs1
ACCESS policy SUCCESS V-288-0 Policy policy1 for fs fs1 deleted
successfully.
```

8 Rename policy2 to policy3.

```
Storage> fs policy rename policy2 policy3
ACCESS policy SUCCESS V-288-0 Policy policy2 renamed to policy3.
```

9 Show the status of policy run for the policy Policy1.

```
Storage> fs policy status Policy1
Policy Name: Policy1
=====
Policy Run Type: normal
Policy Run Status: running
Total Data (Files): 93.1 GB (100000)
Moved/Deleted Data (Files): 47.7 MB (879)
Last File Visited: file100.txt
```

10 Abort the currently running policy Policy1.

```
Storage> fs policy abort Policy1
ACCESS policy INFO V-288-0 Policy Policy1 aborted successfully.
```

11 Start a dry run of the policy Policy1.

```
Storage> fs policy dryrun Policy1
ACCESS policy INFO V-288-0 Policy Policy1 dryrun started in background,
please check 'fs policy status' for progress.
```

12 Pause the currently running policy Policy1.

```
Storage> fs policy pause Policy1
ACCESS policy INFO V-288-0 Policy Policy1 paused successfully.
```


13 Run the currently paused policy Policy1.

```
Storage> fs policy run Policy1
Policy Policy1 is not running currently, as it was killed/paused.
Would you like to start new run (y/n): y
ACCESS policy INFO V-288-0 Policy Policy1 run started in background,
please check 'fs policy status' for progress.
```

14 Resume the currently paused policy Policy1.

```
Storage> fs policy resume Policy1
ACCESS policy INFO V-288-0 Policy Policy1 resume started in background,
please check 'fs policy status' for progress.
```

Obtaining statistics on data usage in the cloud tier in scale-out file systems

You can find the following information for data stored in the cloud tier in a scale-out file system:

- Storage utilization in the cloud
- Number of the objects that are stored in the cloud
- Number of the files that are stored in the cloud
- Number of GET, PUT, and DELETE requests

See the `storage_tier` man page for detailed examples.

To display the number of GET, PUT, and DELETE requests

- ◆ Show the number of GET (read requests), PUT (update or replacement requests), and DELETE (deletion requests).

```
Storage> tier stats show fs_name
          tier_name
```

These statistics are maintained in memory and so are not persistent across reboots.

Example:

```
Storage> tier stats show largefs1 cloudtier
GET          168
GET bytes 174.5MB
PUT          918
PUT bytes 10.3GB
DELETE       20
```

To monitor the usage of data in the cloud tier

- ◆ Monitor the usage of data in the cloud tier.

```
Storage> tier stats monitor fs_name
          tier_name [interval]
```

Example:

```
Storage> tier stats monitor largefs1 cloudtier
GET      GET bytes      PUT      PUT bytes      DELETE
6        384.0MB        4        256.0MB        6
0         0             0         0             0
0         0             0         0             0
0         0             2        128.0MB        0
0         0             0         0             0
```

The default interval is five seconds.

To show the total data usage in the cloud tier

- ◆ Show the total data usage in the cloud tier.

```
Storage> tier stats usage fs_name
            tier_name
```

Unlike the `Storage> tier stats show` command, these statistics are persistent across reboots.

Example:

```
Storage> tier stats usage largefs1 cloudtier
Storage Utilized      :223.1GB
Number of objects     :488
Number of files       :231
```

This example shows that 223.1 GB is used in the cloud tier. Based on the size of the file, each file is chunked to multiple objects when moved to the cloud, so 231 files were stored as 488 objects in the cloud.

To reset the in-memory statistics of the cloud tier to zero

- ◆ Reset the statistics of the specified cloud tier to zero.

```
Storage> tier stats reset fs_name tier_name
```

After executing the `Storage> tier stats reset` command, the output for the `Storage> tier stats show` command is reset to zero.

Provisioning and managing Veritas Access shares

- [Chapter 18. Creating shares for applications](#)
- [Chapter 19. Creating and maintaining NFS shares](#)
- [Chapter 20. Creating and maintaining CIFS shares](#)
- [Chapter 21. Using Veritas Access with OpenStack](#)

Creating shares for applications

This chapter includes the following topics:

- [About file sharing protocols](#)
- [About concurrent access](#)
- [Sharing directories using CIFS and NFS protocols](#)
- [Sharing a file system as a CIFS home directory](#)
- [About concurrent access with NFS and S3](#)

About file sharing protocols

Veritas Access provides support for multiple file sharing protocols.

Veritas Access offers unified access, which provides the option to share a file system or a directory in a file system with more than one protocol. For unified access, only certain protocols combinations are supported.

See [“About concurrent access”](#) on page 270.

Table 18-1 Protocols

Protocol	Definition
Amazon S3	The object server lets you store and retrieve the data that is stored in Veritas Access using the Amazon Simple Storage Service (Amazon S3) protocol. See “About the Object Store server” on page 187.

Table 18-1 Protocols (*continued*)

Protocol	Definition
CIFS	<p>CIFS is active on all nodes within the Veritas Access cluster. The specific shares are read/write on the node they reside on, but can failover to any other node in the cluster. Veritas Access supports CIFS home directory shares.</p> <p>See “About configuring Veritas Access for CIFS” on page 119.</p>
FTP	<p>Allows clients to access files on Veritas Access servers.</p> <p>See “About FTP” on page 174.</p>
NFS	<p>All the nodes in the cluster can serve the same NFS share at the same time in read-write mode. This creates very high aggregated throughput rates, because you can use the sum of the bandwidth of all the nodes. Cache-coherency is maintained throughout the cluster.</p> <p>Veritas Access supports both the NFS kernel-based server and the NFS-Ganesha server in a mutually exclusive way.</p> <p>See “About using NFS server with Veritas Access” on page 108.</p>
Oracle Direct NFS	<p>Optimized NFS client that provides faster access to NFS storage located on NAS storage devices.</p> <p>See “About using Veritas Access with Oracle Direct NFS” on page 166.</p>

About concurrent access

Veritas Access provides support for multi-protocol file sharing where the same file system can be exported to both Windows and UNIX users using the Common Internet File System (CIFS), Network File System (NFS), and Simple Storage Service (S3) protocols. The result is an efficient use of storage by sharing a single data set across multiple application platforms.

Note: When a share is exported over both NFS and CIFS protocols, the applications running on the NFS and CIFS clients may attempt to concurrently read or write the same file. This may lead to unexpected results, such as reading stale data, since the locking models used by these protocols are different. For this reason, Veritas Access warns you when the share export is requested over NFS or CIFS and the same share has already been exported for write access over CIFS or NFS.

The following sections describe concurrent access with multiple protocols.

See [“Sharing directories using CIFS and NFS protocols”](#) on page 271.

See [“About concurrent access with NFS and S3”](#) on page 274.

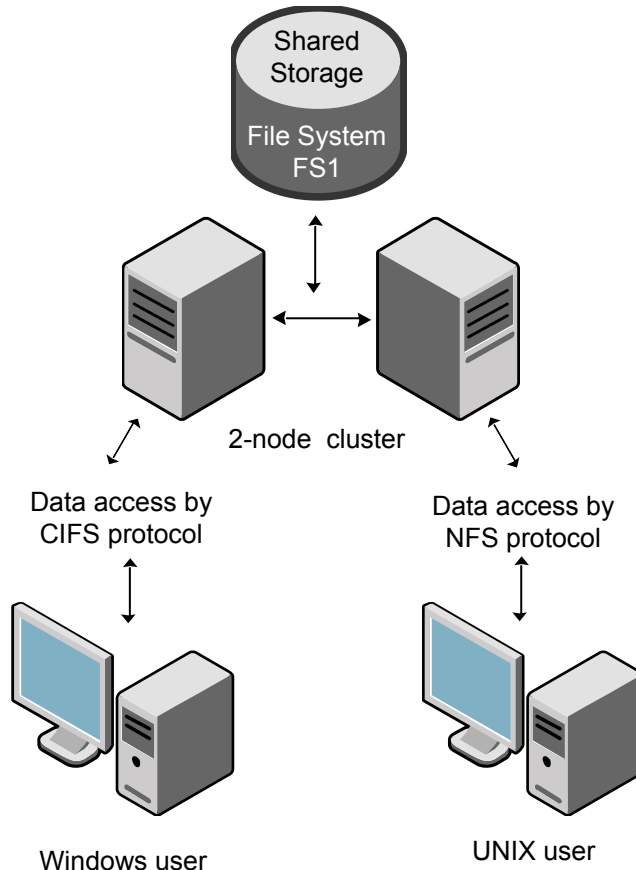
See [“About file sharing protocols”](#) on page 269.

Sharing directories using CIFS and NFS protocols

Veritas Access provides support for multi-protocol file sharing where the same directory or file system can be exported to both Windows and UNIX users using the CIFS and NFS protocols. The result is an efficient use of storage by sharing a single data set across multi-application platforms.

[Figure 18-1](#) shows how the directory sharing for the two protocols works.

Figure 18-1 Exporting and/or sharing CIFS and NFS directories



It is recommended that you disable the `oplocks` option when the following occurs:

- A file system is exported over both the CIFS and NFS protocols.
- Either the CIFS and NFS protocol is set with read and write permission.

To export a directory to Windows and UNIX users

- 1 To export a directory to Windows and UNIX users with read-only and read-write permission respectively, enter the `CIFS` mode and enter the following commands:

```
CIFS> show

          Name  Value
        ----  -
netbios name  Pei60
ntlm auth     yes
allow trusted domains no
homedirfs
aio size      0
idmap backend  rid:10000-1000000
workgroup     PEI-DOMAIN
security      ads
Domain        PEI-DOMAIN.COM
Domain user   Administrator
Domain Controller 10.200.107.251
Clustering Mode normal

CIFS> share add fs1 share1 ro
Exporting CIFS filesystem : share1...
CIFS> share show
ShareName FileSystem ShareOptions
share1 fs1 owner=root,group=root,ro
```

Exit CIFS mode:

```
CIFS> exit
```

- 2 Enter the `NFS` mode and enter the following commands:

```
NFS> share add rw fs1
ACCESS nfs WARNING V-288-0 Filesystem (fs1)
is already shared over CIFS with 'ro' permission.
Do you want to proceed (y/n): y
Exporting */vx/fs1 with options rw
..Success.
NFS> share show
/vx/fs1 * (rw)
```

Sharing a file system as a CIFS home directory

When the file system in CIFS is set to `homedirfs`, the Veritas Access software assumes that the file system is exported to CIFS users in read and write mode. Veritas Access does not allow you to export the same file system as an CIFS share and a home directory file system (`homedirfs`). For example, if the file system `fs1` is already exported as a CIFS share, then you cannot set it as `homedirfs`.

To export a file system set as `homedirfs`

- ◆ To request that a file system be used for home directories, you need to export the file system. Go to the `CIFS` mode and enter the following:

```
CIFS> share show
ShareName      FileSystem    ShareOptions
share1         fs1          owner=root,group=root,rw
CIFS> set homedirfs fs1
ACCESS cifs ERROR V-288-615 Filesystem (fs1) is already exported
by another CIFS share.
```

About concurrent access with NFS and S3

Veritas Access supports concurrent access to a shared file system or a directory from both NFS and S3. The supported configurations are:

- Applications or users write data to NFS shares, while other applications or users read the data over S3.
- Applications or users write data to S3 shares, while other applications or users read the data over NFS.

Creating and maintaining NFS shares

This chapter includes the following topics:

- [About NFS file sharing](#)
- [Displaying file systems and snapshots that can be exported](#)
- [Exporting an NFS share](#)
- [Displaying exported directories](#)
- [About managing NFS shares using netgroups](#)
- [Unexporting a directory or deleting NFS options](#)
- [Exporting an NFS share for Kerberos authentication](#)
- [Mounting an NFS share with Kerberos security from the NFS client](#)
- [Exporting an NFS snapshot](#)

About NFS file sharing

The Network File System (NFS) protocol enables exported directories (including all files under the directory that reside on the exported directory's file system) hosted by an NFS server to be accessed by multiple UNIX and Linux client systems.

Using NFS, a local system can mount and use a disk partition or a file system from a remote system (an NFS server), as if it were local. The Veritas Access NFS server exports a directory, with selected permissions and options, and makes it available to NFS clients.

The selected permissions and options can also be updated, by adding or removing permissions, to restrict or expand the permitted use.

The Veritas Access NFS service is clustered. The NFS clients continuously retry during a failover transition. Even if the TCP connection is broken for a short time, the failover is transparent to NFS clients, and NFS clients regain access transparently as soon as the failover is complete.

See [“About using NFS server with Veritas Access”](#) on page 108.

Displaying file systems and snapshots that can be exported

To display a file system and snapshots that can be exported

- ◆ To display online file systems and the snapshots that can be exported, enter the following:

```
NFS> show fs
```

For example:

```
NFS> show fs
FS/Snapshot
=====
fs1
```

Exporting an NFS share

You can export an NFS share with the specified NFS options that can then be accessed by one or more client systems.

If you add a directory that has already been exported with a different NFS option (rw, ro, async, or secure, for example), Veritas Access provides a warning message saying that the directory has already been exported. Veritas Access updates (overwrite) the old NFS options with the new NFS options.

Directory options appear in parentheses.

If a client was not specified when the `NFS> share add` command was used, then * is displayed as the system to be exported to, indicating that all clients can access the directory.

Directories that have been exported to multiple clients appear as separate entries. Directories that are exported to <world> and other specific clients also appear as separate entries.

For example:

Consider the following set of exported directories where only the client (1.1.1.1) has **read-write** access to directory (fs2), while all other clients have **read** access only.

```
/vx/fs2          * (ro)
/vx/fs2  1.1.1.1 (rw)
```

Exporting the same directory to multiple clients with different permissions is not supported with NFS-Ganesha.

When sharing a directory, Veritas Access does not check whether the client exists or not. If you add a share for an unknown client, then an entry appears in the `NFS> show` command output.

The `NFS> show fs` command displays the list of exportable file systems. If a directory does not exist, the directory is automatically created and exported when you try to export it.

Valid NFS options include the following:

<code>rw</code>	Grants read and write permission to the directory (including all files under the directory that reside on the exported directory's file system). Hosts mounting this directory will be able to make changes to the directory.
<code>ro (Default)</code>	Grants read-only permission to the directory. Hosts mounting this directory will not be able to change it.
<code>sync (Default)</code>	Grants synchronous write access to the directory. Forces the server to perform a disk write before the request is considered complete.
<code>async</code>	Grants asynchronous write access to the directory. Allows the server to write data to the disk when appropriate.
<code>secure (Default)</code>	Grants secure access to the directory. Requires that clients originate from a secure port. A secure port is between 1-1024.
<code>insecure</code>	Grants insecure access to the directory. Permits client requests to originate from unprivileged ports (those above 1024).
<code>secure_locks (Default)</code>	Requires authorization of all locking requests. This option is not supported with NFS-Ganesha.

<code>insecure_locks</code>	Some NFS clients do not send credentials with lock requests, and therefore work incorrectly with <code>secure_locks</code> , in which case you can only lock world-readable files. If you have such clients, either replace them with better ones, or use the <code>insecure_locks</code> option. This option is not supported with NFS-Ganesha.
<code>root_squash</code> (Default)	Prevents the root user on an NFS client from having root privileges on an NFS mount. This effectively "squashes" the power of the remote root user to the lowest local user, preventing remote root users from acting as though they were the root user on the local system.
<code>no_root_squash</code>	Disables the <code>root_squash</code> option. Allows root users on the NFS client to have root privileges on the NFS server.
<code>wdelay</code> (Default)	Causes the NFS server to delay writing to the disk if another write request is imminent. This can improve performance by reducing the number of times the disk must be accessed by separate write commands, reducing write overhead. Note: The <code>wdelay</code> option is deprecated, and is supported for backward-compatibility only. This option is not supported with NFS-Ganesha.
<code>no_wdelay</code>	Disables the <code>wdelay</code> option. The <code>no_wdelay</code> option has no effect if the <code>async</code> option is also set. Note: The <code>no_wdelay</code> option is deprecated, and is supported for backward-compatibility only. Using the <code>no_wdelay</code> option is always effective. This option is not supported with NFS-Ganesha.
<code>subtree_check</code>	Verifies that the requested file is in an exported subdirectory. If this option is turned off, the only verification is that the file is in an exported file system. This option is not supported with NFS-Ganesha.
<code>no_subtree_check</code> (Default)	Sometimes subtree checking can produce problems when a requested file is renamed while the client has the file open. If many such situations are anticipated, it might be better to set <code>no_subtree_check</code> . One such situation might be the export of the home directory. Most other situations are best handled with <code>subtree_check</code> . This option is not supported with NFS-Ganesha.

<code>fsid</code> (Default)	Allows the Veritas Access administrator to associate a specific number as <code>fsid</code> with the share. This option is not supported with NFS-Ganesha.
<code>nordirplus</code>	Allows you to disable a <code>readdirplus</code> remote procedure call (RPC).
<code>sec</code>	Specifies the Kerberos security options for exporting an NFS share. The value can be <code>krb5</code> , <code>krb5i</code> , <code>krb5p</code> , or <code>sys</code> . The <code>sys</code> option does not provide Kerberos authentication. The other options use Kerberos V5 to authenticate users to the NFS server.

Note: With `root_squash`, the root user can access the share, but with 'nobody' permissions.

To export a directory/file system

- 1 To see your exportable online file systems and snapshots, enter the following:

```
NFS> show fs
```

- 2 To see your NFS shares and their options, enter the following:

```
NFS> share show
```

- 3 To export a directory, enter the following command:

```
NFS> share add nfsoptions export_dir [client]
```

`nfsoptions` Comma-separated list of export options from the set.

`export_dir` Specifies the name of the directory you want to export.

The directory name should start with `/vx`, and only `a-zA-Z0-9_/@+.=:-` characters are allowed for `export_dir`.

client

Clients may be specified in the following ways:

- Single host - specify a host either by an abbreviated name that is recognized by the resolver (DNS is the resolver), the fully qualified domain name, or an IP address.
- Netgroups - specify netgroups as *@group*. Only the host part of each netgroup member is considered for checking membership.
- IP networks - specify an IP address and netmask pair (address/netmask) to simultaneously export directories to all hosts on an IP sub-network. Specify the netmask as a contiguous mask length. You can specify either an IPv4 address or an IPv6 address.

If the client is not given, then the specified directory can be mounted or accessed by any client. To re-export new options to an existing share, the new options will be updated after the command is run.

Displaying exported directories

You can display the exported directories and the NFS options that are specified when the directory was exported. For NFS-Ganesha (GNFS), the output also shows the virtual IP address that must be used on the client to mount the GNFS shares for the shares that are exported from 'largefs' file systems.

To display exported directories

To display exported directories, enter the following:

```
NFS> share show
```

The command output displays the following columns:

Left-hand column Displays the directory that was exported.

Right-hand column Displays the system that the directory is exported to, and the NFS options with which the directory was exported.

The command output displays the following columns:

Left-hand column Displays the directory that was exported.

Middle column Displays the system that the directory is exported to, and the NFS options with which the directory was exported.

Right-hand column Displays the virtual IP address that must be used on the client to mount the GNFS shares of scale-out file systems.

About managing NFS shares using netgroups

A netgroup defines a network-wide group of hosts and users. You use netgroups for restricting access to shared NFS file systems and to restrict remote login and shell access.

Each line in the netgroup file consists of a netgroup name followed by a list of members, where a member is either another netgroup name, or a comma-separated list of host, user, or a domain. Host, user, and domain are character strings for the corresponding components. Any of these three fields can be empty, which indicates a wildcard, or may consist of the string "-" to indicate that there is no valid value for the field. The domain field must either be the local domain name or empty for the netgroup entry to be used. This field does not limit the netgroup or provide any security. The domain field refers to the domain in which the host is valid, not the domain containing the trusted host.

When exporting a directory by NFS with the specified options, clients may be specified using netgroups. Netgroups are identified using @group. Only the host part of each netgroup member is considered when checking for membership.

```
NFS> share add rw,async /vx/fs1/share @client_group
```

Unexporting a directory or deleting NFS options

You can unexport the share of the exported directory.

Note: You will receive an error message if you try to remove a directory that does not exist.

To unexport a directory or delete NFS options

- 1 To see your existing exported resources, enter the following command:

```
NFS> share show
```

Only the directories that are displayed can be unexported.

For example:

```
NFS> share show
/vx/fs2          * (sync)
/vx/fs3          * (secure,ro,no_root_squash)
```

- 2 To delete a directory from the export path, enter the following command:

```
NFS> share delete export_dir [client]
```

For example:

```
NFS> share delete /vx/fs3
Removing export path */vx/fs3
..Success.
```

export_dir	<p>Specifies the name of the directory you want to delete.</p> <p>The directory name should start with /vx, and only a-zA-Z0-9_/@+=.:- characters are allowed in export_dir.</p> <p>You cannot include single or double quotes that do not enclose characters.</p>
-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
NFS> share delete "*/vx/example"
```

client

Clients may be specified in the following ways:

- Single host - specify a host either by an abbreviated name that is recognized by the resolver (DNS is the resolver), the fully qualified domain name, or an IP address.
- Netgroups - specify netgroups as @group. Only the host part of each netgroup member is considered for checking membership.
- IP networks - specify an IP address and netmask pair (address/netmask) to simultaneously export directories to all hosts on an IP sub-network. Specify the netmask as a contiguous mask length.

If *client* is included, the directory is removed from the export path that was directed at the *client*.

If a directory is being exported to a specific client, the `NFS> share delete` command must specify the client to remove that export path.

If the client is not specified, then the specified directory can be mounted or accessed by any client.

Exporting an NFS share for Kerberos authentication

Kerberos provides three types of security options for exporting an NFS share:

- `krb5`
- `krb5i`
- `krb5p`

Veritas Access also provides a `sys (sec=sys)` export option, which does not provide Kerberos authentication. Veritas Access supports all of the three types of Kerberos security options. All of the security options use Kerberos V5 to authenticate users to NFS servers.

`krb5i` computes a hash on every remote procedure (RPC) call request to the server and every response to the client. The hash is computed on an entire message: RPC header, plus NFS arguments or results. Since the hash information travels with the NFS packet, any attacker modifying the data in the packet can be detected. Thus `krb5i` provides integrity protection.

`krb5p` uses encryption to provide privacy. With `krb5p`, NFS arguments and results are encrypted, so a malicious attacker cannot spoof on the NFS packets and see file data or metadata.

Note: Since `krb5i` and `krb5p` perform an additional set of computations on each NFS packet, NFS performance decreases as compared with `krb5`.

Performance decreases in the following order: `krb5` > `krb5i` > `krb5p`.

`krb5` provides better performance and `krb5p` gives the least performance.

Additional export options are available.

See “[Exporting an NFS share](#)” on page 276.

To export a directory using only the `krb5` mount option

- ◆ Export a directory using only the `krb5` mount option:

```
NFS> share add sec=krb5 /vx/fs1
Exporting /vx/fs1 with options sec=krb5
Success.
```

To export a directory using `krb5`, `krb5i`, `krb5p`, and `sys` options

- ◆ Export a directory using `krb5`, `krb5i`, `krb5p`, and `sys` options.

```
NFS> share add sec=krb5:krb5i:krb5p:sys /vx/fs1
Exporting /vx/fs1 with options sec=krb5:krb5i:krb5p:sys
Success.
```

Different clients can use different levels of security in this case. Client A can mount with `krb5`, and client B can mount with `krb5p`. If no mount option is given at the client side, security to be chosen is negotiated, and the highest level of security is chosen. In this case, it is `krb5p`.

Mounting an NFS share with Kerberos security from the NFS client

This section explains how the NFS client will NFS mount with the Kerberos mount options. This procedure assumes that the NFS service principal of the NFS client is added to the KDC server, and the keytab is copied at the appropriate location on the client.

The steps may differ depending on the operating system and version of the client. On a Red Hat Enterprise Linux (RHEL) client, Kerberos can be configured as follows.

To mount the NFS client with the Kerberos mount options

- 1 Create the NFS service principal for the client on the KDC server and copy it to the client system at `/etc/krb5.keytab`.
- 2 Configure the `/etc/krb5.conf` file with the KDC details.
- 3 Enable `SECURE_NFS=yes` in the `/etc/sysconfig/nfs` file.
- 4 Start the `rpcgssd` service.

```
# service rpcgssd start
```

- 5 Keep the clocks of the KDC server, the Veritas Access server, and the NFS client in sync.

A maximum of a five-minute variation is accepted, or otherwise the Kerberos NFS mount fails.

```
[root@krb-client]# mount -o vers=4,sec=krb5 10.209.107.24:/vx/fs2/share1 /mnt/share1
```

Make sure that the virtual IP that is used for mounting can use reverse name lookup to the Veritas Access cluster name. For example, if `access_ga` is the cluster name, then in the above example, `access_ga` should look up to 10.209.107.24 and vice versa. If the IP 10.209.107.24 can be looked up by multiple host names, make sure that the entry `access_ga` is first in the reverse lookup.

- 6 Make sure the users accessing the NFS share are already added on the KDC server.

Use `kinit` to get the ticket granting ticket from the KDC server on the NFS client.

```
[root@krb-client]# su - sfuuser2

[sfuuser2@krb-client ~]$ kinit
Password for sfuuser2@TESTKDC.COM:
[sfuuser2@krb-client ~]$ cd /mnt/share1
[sfuuser2@krb-client share1]$ touch test.txt
[sfuuser2@krb-client share1]$
[sfuuser2@krb-client share1]$ ls -al total 4
drwxrwxrwx 2 root root 96 May 14 16:03 .
drwxr-xr-x. 17 root root 4096 May 7 19:41 ..
-rw-r--r-- 1 sfuuser2 sfugroup1 0 May 14 16:03 test.txt
```

Exporting an NFS snapshot

To export an NFS snapshot

- 1 For example, to create an NFS snapshot, enter the following:

```
Storage> snapshot create fs5spl FS5
```

See [“About snapshots”](#) on page 435.

- 2 For example, to export the NFS snapshot, enter the following:

```
NFS> share add rw /vx/FS5:fs5spl
```

See [“Exporting an NFS share ”](#) on page 276.

Creating and maintaining CIFS shares

This chapter includes the following topics:

- [About managing CIFS shares](#)
- [Exporting a directory as a CIFS share](#)
- [Configuring a CIFS share as secondary storage for an Enterprise Vault store](#)
- [Exporting the same file system/directory as a different CIFS share](#)
- [About the CIFS export options](#)
- [Setting share properties](#)
- [Hiding system files when adding a CIFS normal share](#)
- [Displaying CIFS share properties](#)
- [Allowing specified users and groups access to the CIFS share](#)
- [Denying specified users and groups access to the CIFS share](#)
- [Exporting a CIFS snapshot](#)
- [Deleting a CIFS share](#)
- [Modifying a CIFS share](#)
- [Making a CIFS share shadow copy aware](#)
- [Creating CIFS shares for a scale-out file system](#)

About managing CIFS shares

You can export the Veritas Access file systems to clients as CIFS shares. When a share is created, it is given a name. The share name is different from the file system name. Clients use the share name when they import the share.

You create and export a share with one command. The same command binds the share to a file system, and you can also use it to specify share properties.

In addition to exporting file systems as CIFS shares, you can use Veritas Access to store user home directories. Each of these home directories is called a home directory share. Shares that are used to export ordinary file systems (that is, file systems that are not used for home directories), are called ordinary shares to distinguish them from home directory shares.

Exporting a directory as a CIFS share

Directory-level share support is available only in the ctdb clustering mode. If you want to export a directory as a CIFS share, you must first switch to the ctdb clustering mode.

See [“About CIFS clustering modes”](#) on page 159.

To check the status of the CIFS server to confirm that the clustering mode is set to ctdb

- ◆ To check the status of the CIFS server to confirm that the clustering mode is set to ctdb, enter the following:

```
CIFS> server status
```

To export a directory as a CIFS share

- 1 To export a directory as a CIFS share, enter the following:

```
CIFS> share add fs1/access share1 rw,full_acl
```

If the directory name contains a space, enter the directory name with double quotes (" ").

- 2 To list the CIFS shares, enter the following:

```
CIFS> share show
```


Configuring a CIFS share as secondary storage for an Enterprise Vault store

You can use Veritas Access as secondary storage with Enterprise Vault 12.0 by exporting the file system over the CIFS protocol.

Note: Before configuring the CIFS share path as secondary storage, you need to verify that the CIFS share path is accessible. Confirm that I/O operations can occur on the CIFS share.

Configuring a CIFS share as secondary storage for an Enterprise Vault store

- 1 On the Veritas Access cluster, you export the file system over the CIFS protocol using the following CIFS export options: `fs_mode=1777,rw,full_acl`.
 See [“About the CIFS export options”](#) on page 290.
- 2 On the Enterprise Vault server, open the Enterprise Vault console.
- 3 Right-click on the partition that is created on **Vault Store > Properties**.
 Enterprise Vault brings up the **Vault Store Partition Properties** window.
- 4 In the **Vault Store Partition Properties** window, select the **Migration** tab.
- 5 Specify the path of the CIFS share in the **Secondary storage location** text box.

Example:

`\\IP address of the CIFS share\name of file system`

- 6 Press **Apply**.

Exporting the same file system/directory as a different CIFS share

In ctdb clustering mode, you can export the same file system or directory as a different CIFS share with different available CIFS options. This feature allows you more granular control over CIFS shares for different sets of users.

If the same file system is exported as different shares in ctdb clustering mode, then after switching to normal clustering mode only one share out of these is available.

Note: If the same file system or directory is exported as different shares, then the `fs_mode` value is the same for all of these shares; that is, the last modified `fs_mode` value is applicable for all of those shares.

Note: This feature is only supported in the ctdb clustering mode.

To export a directory with read access to everyone, but write access to the limited set of users who need to be authenticated

- ◆ To export a directory with read access to everyone, but write access to the limited set of users who need to be authenticated, enter the following:

```
CIFS> share add "fsl/Veritas isa" share1 rw,noguest
CIFS> share add "fsl/Veritas isa" share2 ro,guest
CIFS> share show
```

The above example illustrates that the same directory is exported as a different CIFS share for `guest` and `noguest` users with different sets of permissions.

About the CIFS export options

The following are the CIFS export options.

Table 20-1 CIFS export options

CIFS export option	Definition
rw	<p>There is a share option which specifies if the files in the share will be read-only or if both read and write access will be possible, subject to the authentication and authorization checks when a specific access is attempted. This share option can be given one of these values, either <code>rw</code> or <code>ro</code>.</p> <p>Grants read and write permission to the exported share.</p>
ro (Default)	<p>Grants read-only permission to the exported share. Files cannot be created or modified.</p>

Table 20-1 CIFS export options (*continued*)

CIFS export option	Definition
guest	<p>Another configuration option specifies if a user trying to establish a CIFS connection with the share must always provide the user name and password, or if they can connect without it. In this case, only restricted access to the share will be allowed. The same kind of access is allowed to <code>anonymous</code> or <code>guest</code> user accounts. This share option can have one of the following values, either <code>guest</code> or <code>noguest</code>.</p> <p>Veritas Access allows restricted access to the share when no user name or password is provided.</p>
noguest (Default)	Veritas Access always requires the user name and password for all of the connections to this share.
full_acl	All Windows Access Control Lists (ACLs) are supported except in the case when you attempt using the Windows Explorer folder Properties > Security GUI to inherit down to a non-empty directory hierarchy while denying all access to yourself.
no_full_acl (Default)	Some advanced Windows Access Control Lists (ACLs) functionality does not work. For example, if you try to create ACL rules on files saved in a CIFS share using Windows explorer while allowing some set of file access for <code>user1</code> and denying file access for <code>user2</code> , this is not possible when CIFS shares are exported using <code>no_full_acl</code> .
hide_unreadable	<p>Prevents clients from seeing the existence of files and directories that are not readable to them.</p> <p>The default is: <code>hide_unreadable</code> is set to off.</p>
veto_sys_files	To hide some system files (lost+found, quotas, quotas.grp) from displaying when using a CIFS normal share, you can use the <code>veto_sys_files</code> CIFS export option. For example, when adding a CIFS normal share, the default is to display the system files. To hide the system files, you must use the <code>veto_sys_files</code> CIFS export option.
fs_mode	<p>When a file system or directory is exported by CIFS, its mode is set to an <code>fs_mode</code> value. It is the UNIX access control set on a file system, and CIFS options like <code>rw/ro</code> do not take precedence over it. This value is reset to <code>0755</code> when the CIFS share is deleted.</p> <p>The default is: <code>fs_mode = 1777</code>.</p>

Table 20-1 CIFS export options (*continued*)

CIFS export option	Definition
dir_mask	<p>When a directory is created under a file system or directory exported by CIFS, the necessary permissions are calculated by mapping DOS modes to UNIX permissions. The resulting UNIX mode is then bit-wise 'AND'ed with this parameter. Any bit not set here is removed from the modes set on a directory when it is created.</p> <p>The default is: <code>dir_mask = 0775</code>.</p>
create_mask	<p>When a file is created under a file system or directory exported by CIFS, the necessary permissions are calculated by mapping DOS modes to UNIX permissions. The resulting UNIX mode is then bit-wise 'AND'ed with this parameter. Any bit not set here is removed from the modes set on a file when it is created.</p> <p>The default is: <code>create_mask = 0775</code>.</p>
oplocks (Default)	<p>Veritas Access supports the CIFS opportunistic locks. You can enable or disable them for a specific share. The opportunistic locks improve performance for some workloads, and there is a share configuration option which can be given one of the following values, either oplocks or nooplocks.</p> <p>Veritas Access supports opportunistic locks on the files in this share.</p>
nooplocks	<p>No opportunistic locks will be used for this share.</p> <p>Disable the oplocks when:</p> <ul style="list-style-type: none"> ■ 1) A file system is exported over both CIFS and NFS protocols. ■ 2) Either CIFS or NFS protocol has read and write access.
owner	<p>There are more share configuration options that can be used to specify the user and group who own the share. If you do not specify these options for a share, Veritas Access uses the current values as default values for these options. You may want to change the default values to allow a specific user or group to be the share owner.</p> <p>Irrespective of who are owner and group of the exported share, any CIFS clients can create folders and files in the share. However, there are some operations that require owner privileges; for example, changing the owner itself, and changing permissions of the top-level folder (that is, the root directory in UNIX terms). To enable these operations, you can set the owner option to a specific user name, and this user can perform the privileged operations.</p>

Table 20-1 CIFS export options (*continued*)

CIFS export option	Definition
group	By default, the current group is the primary group owner of the root directory of the exported share. This lets CIFS clients create folders and files in the share. However, there are some operations that require group privileges; for example, changing the group itself, and changing permissions of the top-level folder (that is, the root directory in UNIX terms). To enable these operations, you can set the <code>group</code> option to a specific group name, and this group can perform the privileged operations.
ip	<p>Veritas Access lets you specify a virtual IP address. If you set <code>ip=virtualip</code>, the share is located on the specified virtual IP address. This address must be part of the Veritas Access cluster, and is used by the system to serve the share internally.</p> <p>Note: <code>ip</code> is not a valid CIFS option when using the <code>ctdb</code> clustering mode.</p> <p>See “About CIFS clustering modes” on page 159.</p>
max_connections	<p>Specify the maximum limit for concurrent CIFS connections for a CIFS share.</p> <p>The default value is 0, indicating that there are no limited connections.</p>
shadow_copy	<p>Indicates that this is a <code>shadow_copy</code> capable CIFS share.</p> <p>See “Making a CIFS share shadow copy aware” on page 300.</p>
enable_encryption	<p>If <code>enable_encryption</code> is set, then all the traffic to a share must be encrypted once the connection has been made to the share. The server will return an <code>access denied</code> message to all unencrypted requests on such a share. As SMB3 is the max protocol, only SMB3 clients supporting encryption will be able to connect to the share.</p>
disable_encryption	<p>If <code>disable_encryption</code> is set, then encryption cannot be negotiated by the client. SMB1, SMB2, and SMB3 clients can connect to the share.</p>
enable_durable_handles	<p>Enables support for durable handles for CIFS shares. Enabling this option disables use of POSIX/fcntl locks. Exporting the same CIFS share using NFS may result in data corruption. For support for durable handles on CIFS shares, you must specify this option.</p>

Setting share properties

After a file system is exported as a CIFS share, you can change one or more share options. This is done using the same `share add` command, giving the name of an existing share and the name of the file system exported with this share. Veritas Access will realize the given share has already been exported and that it is only required to change the values of the share options.

For example, to export the file system `fs1` with the name `share1`, enter the following:

```
CIFS> share add fs1 share1 "owner=administrator,group=domain users,rw"
CIFS> share show
```

To export a file system

- ◆ Export a file system, enter the following:

```
CIFS> share add filesystem sharename \
[@virtual_ip] [cifsoptions]
```

filesystem A Veritas Access file system that you want to export as a CIFS share. The given file system must not be currently used for storing the home directory shares.

The file system or directory path should always start with the file system name, not with the file system mount point `/vx`.

sharename The name for the newly-exported share. Names of the Veritas Access shares can consist of the following characters: lower and uppercase letters "a" - "z" and "A" - "Z," numbers "0" - "9" and special characters: "_" and "-". ("-" cannot be used as the first character in a share name).

Note: A share name cannot exceed 256 characters.

@virtual_ip Specifies an optional full identifier allowing a virtual IP to access the specified CIFS share.

Veritas Access provides unified access to all shares through virtual IPs.

If a CIFS share is added with the `@virtual_ip` full identifier, the CIFS share is created by allowing only this virtual IP to access this CIFS share.

```
CIFS> share show
```

cifsoptions A comma-separated list of CIFS export options. This part of the command is optional.

If a CIFS export option is not provided, Veritas Access uses the default value.

See [“About the CIFS export options”](#) on page 290.

For example, an existing file system called `FSA` being exported as a share called `ABC`:

```
CIFS> share add FSA ABC rw,guest,owner=john,group=abcdev
```

Hiding system files when adding a CIFS normal share

When adding a CIFS normal share, the default is to display the system files (`lost+found`, `quotas`, `quotas.grp`). To hide the system files, you must use the `veto_sys_files` CIFS export option.

See [“About the CIFS export options”](#) on page 290.

To hide system files when adding a CIFS normal share

- ◆ To hide system files when adding a CIFS normal share, enter the following:

```
CIFS> share add filesystem
      sharename [cifsoption]
```

Use the `veto_sys_files` CIFS export option to hide system files.

Displaying CIFS share properties

To display share properties

- 1 To display the information about all of the exported shares, enter the following:

```
CIFS> share show
```

- 2 To display the information about one specific share, enter the following:

```
CIFS> share show sharename
```

Allowing specified users and groups access to the CIFS share

To allow specified users and groups access to the CIFS share

- ◆ To allow specified users and groups access to the CIFS share, enter the following:

```
CIFS> share allow sharename
      @group1 \
      [, @group2, user1, user2, ...]
```

sharename	<p>Name of the CIFS share for which you want to allow specified users and groups access.</p> <p>Names of the Veritas Access shares are non case sensitive and can consist of the following characters: lower and uppercase letters "a" - "z" and "A" - "Z," numbers "0" - "9" and special characters: "_" and "-". ("-", cannot be used as the first character in a share name).</p>
group	<p>If the CIFS server joined a domain, and there is a space in the user or group name, the user or group name needs to be entered with double quotes (for example, "@domain users").</p> <p>By default, all groups are allowed to access the shares.</p> <p>In the case where a CIFS share has joined a domain, and the domain contains trusted domains, and <code>allow_trusted_domains</code> is set to <code>yes</code> on the CIFS server, if you want to allow/deny users or groups from the trusted domains, the user or group needs to be prefixed with the trusted domain name. Separate the domain and user/group with a double backslash.</p> <p>For example:</p> <pre>CIFS> share allow sharename "@domain name\\group name"</pre>
user	<p>Name of the CIFS user allowed access to the CIFS share.</p> <p>By default, all users are allowed to access the shares.</p>

If `all` is specified, then default access restrictions are restored on the CIFS share.

```
CIFS> share allow share1 user1, @group1
```


Denying specified users and groups access to the CIFS share

To deny specified users and groups access to the CIFS share

- ◆ To deny specified users and groups access to the CIFS share, enter the following:

```
CIFS> share deny sharename \  
@group1[,@group2,user1,user2,...]
```

sharename Name of the CIFS share for which you want to deny specified users and groups access.

Names of the Veritas Access shares are non case sensitive and can consist of the following characters: lower and uppercase letters "a" - "z" and "A" - "Z," numbers "0" - "9" and special characters: "_" and "-". ("-", cannot be used as the first character in a share name).

group If the CIFS server joined a domain, and there is a space in the user or group name, the user or group name needs to be entered with double quotes (for example, "@domain users").

By default, all groups are allowed to access the shares.

In the case where a CIFS share has joined a domain, and the domain contains trusted domains, and CIFS is set to trusted domains as true, if you want to allow/deny users or groups from the trusted domains, the user or group needs to be prefixed with the trusted domain name. Separate the domain and user/group with a double backslash.

For example:

```
CIFS> share deny sharename  
"@domain name\\user name"
```

user Name of the CIFS user denied access to the CIFS share.

By default, all users are allowed to access the shares.

If *all* is specified, then all the users and groups are not able to access the share.

```
CIFS> share deny share1 user1,@group1
```

Exporting a CIFS snapshot

To export a CIFS snapshot

- 1 To create a CIFS snapshot, enter the following for example:

```
Storage> snapshot create cf11sp1 CF11
```

See “[About snapshots](#)” on page 435.

- 2 To export the CIFS snapshot, enter the following for example:

```
CIFS> share add CF11:cf11sp1 cf11sp1 rw,guest
```

A client can access the CIFS snapshot by the CIFS share name, `cf11sp1`.

Deleting a CIFS share

To delete a CIFS share

- 1 To delete a share, enter the following:

```
CIFS> share delete sharename [@virtual_ip]
```

sharename Specifies the name of the share that you want to delete.

@virtual_ip Specifies an optional full identifier allowing a virtual IP to access the specified CIFS share.

For example:

```
CIFS> share delete share1
```

- 2 To confirm the share is no longer exported, enter the following:

```
CIFS> share show
```

In the case of any remanent sessions (sessions that are not closed while deleting a CIFS share), Veritas Access displays the following output:

```
CIFS> share delete share2
```

This is a rare situation, and it occurs if the following conditions are met:

- CIFS server is online
- CIFS share that is being deleted is ONLINE
- There are some existing client connections with that CIFS share

- While deleting the share, some remanent sessions are left

If any condition is failed above, then the `CIFS> share delete` command output displays as usual.

```
CIFS> share delete share2
```

Modifying a CIFS share

You can re-export the file system with the given share name. The new options are updated after the command is run.

To modify a CIFS share

- ◆ To modify a CIFS share, enter the following:

```
CIFS> share modify sharename[@virtual_ip] [cifsoptions]
```

sharename	<p>The name of the CIFS share that you want to modify.</p> <p>Names of the Veritas Access shares can consist of the following characters: lower and uppercase letters "a" - "z" and "A" - "Z," numbers "0" - "9" and special characters: "_" and "-". ("-" cannot be used as the first character in a share name).</p>
@virtual_ip	<p>Specifies an optional full identifier allowing a virtual IP to access the specified CIFS share.</p> <p>Veritas Access provides unified access to all shares through virtual IPs.</p>
cifsoptions	<p>A comma-separated list of CIFS export options.</p> <p>If a CIFS export option is not provided, Veritas Access uses the default value.</p> <p>See "About the CIFS export options" on page 290.</p>

For example:

```
CIFS> share modify share2 ro,full_acl
```

```
CIFS> share show
```

Making a CIFS share shadow copy aware

Shadow Copy (Volume Snapshot Service or Volume Shadow Copy Service or VSS) is a technology included in Microsoft Windows that allows taking manual or automatic backup copies or snapshots of data on a specific volume at a specific point in time over regular intervals.

To make a CIFS share shadow copy aware

- ◆ Add the CIFS export option `shadow_copy` to the CIFS share.

For example:

```
CIFS> share add fs1 share1 rw,shadow_copy
```

```
CIFS> share show share1
```

See [“About the CIFS export options”](#) on page 290.

Creating CIFS shares for a scale-out file system

Veritas Access provides support for creating CIFS shares for a scale-out file system. Unlike a standard clustered file system, a scale-out file system can scale linearly and you can grow the file system up to 3 PB.

See [“About scale-out file systems”](#) on page 211.

The following limitations apply when you create a CIFS share for a scale-out file system:

- In CTDB mode:
CTDB mode is not supported for a scale-out filesystem.
 - If the CIFS server is in normal mode, you can add shares to both clustered file system and scale-out file system.
 - If the CIFS server is in CTDB mode, you can add shares to clustered file system but you cannot add shares to a scale-out file system.
 - If clustered file system shares as well as scale-out file system shares exist, you cannot switch the CIFS server from normal mode to CTDB mode.
- When setting the home directory file systems:
 - You cannot have a home directory file system which is a combination of scale-out file system and clustered file system.
 - You can set multiple clustered file systems as a home directory file system.
- When setting quotas:

The `CIFS> homedir set` command is usually used along with the `Storage> quota` functionality. Since a scale-out file system does not support quota, a CIFS home directory which is based on a scale-out file system does not have the quota functionality.

Using Veritas Access with OpenStack

This chapter includes the following topics:

- [About the Veritas Access integration with OpenStack](#)
- [About the Veritas Access integration with OpenStack Cinder](#)
- [About the Veritas Access integration with OpenStack Manila](#)

About the Veritas Access integration with OpenStack

OpenStack is a cloud operating system that controls large pools of computer, storage, and networking resources in a data center. OpenStack provides a dashboard that lets you provision resources using a web interface.

Veritas Access is integrated with the following OpenStack components:

- Cinder - is a block storage service for OpenStack. Cinder provides the infrastructure for managing volumes in OpenStack. Cinder volumes provide persistent storage to guest virtual machines (known as instances) that manage OpenStack compute software. Cinder allows the ability for OpenStack instances to use the storage hosted by Veritas Access.
See [“About the Veritas Access integration with OpenStack Cinder”](#) on page 303.
- Manila - lets you share Veritas Access file systems with virtual machines on OpenStack.
See [“About the Veritas Access integration with OpenStack Manila”](#) on page 312.

About the Veritas Access integration with OpenStack Cinder

Cinder is a block storage service for OpenStack. Cinder provides the infrastructure for managing volumes in OpenStack. Cinder volumes provide persistent storage to guest virtual machines (known as instances) that manage OpenStack compute software.

Veritas Access is integrated with OpenStack Cinder, which provides the ability for OpenStack instances to use the storage hosted by Veritas Access.

Table 21-1 Mapping of OpenStack Cinder operations to Veritas Access

Operation in OpenStack Cinder	Operation in Veritas Access
Create and delete volumes	Create and delete files.
Attach and detach the volumes to virtual machines	This operation occurs on the OpenStack controller node. This operation is not applicable in Veritas Access.
Create and delete snapshots of the volumes	Create and delete the snapshot files of the volume.
Create a volume from a snapshot	This operation occurs on the OpenStack controller node. This operation is not applicable in Veritas Access.
Copy images to volumes	This operation occurs on the OpenStack controller node. This operation is not applicable in Veritas Access.
Copy volumes to images	This operation occurs on the OpenStack controller node. This operation is not applicable in Veritas Access.
Extend volumes	Extending files.

Note: To perform these operations, you need to use the OpenStack Cinder commands, not the Veritas Access commands.

The Veritas NFS OpenStack Cinder driver is a Python script that is checked in to the OpenStack source code in the public domain. To use the Veritas Access

integration with OpenStack Cinder, you need to make some configuration changes on the OpenStack controller node.

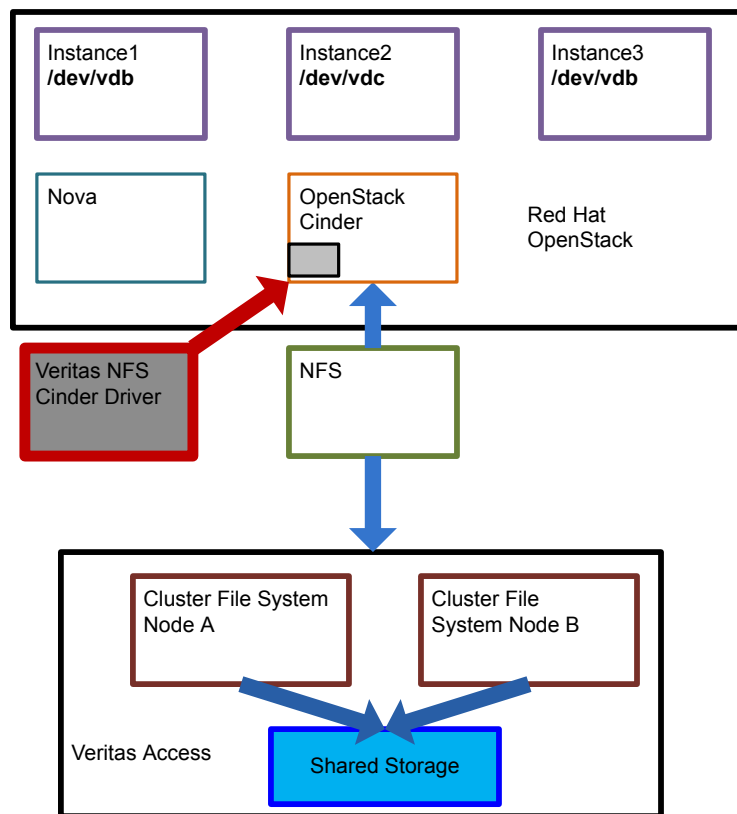
For the supported OpenStack versions for running the OpenStack Cinder driver, see the *Veritas Access Installation Guide*.

About the Veritas Access integration with OpenStack Cinder architecture

Figure 21-1 describes the Veritas Access integration with OpenStack Cinder architecture.

OpenStack instances are the individual virtual machines running on physical compute nodes. The compute service, Nova, manages the OpenStack instances. The compute service, Nova, manages the OpenStack instances.

Figure 21-1 Veritas Access integration with OpenStack Cinder architecture



Configuring Veritas Access with OpenStack Cinder

To show all your NFS shares

- ◆ To show all your NFS shares that are exported from Veritas Access, enter the following:

```
OPENSTACK> cinder share show
```

For example:

```
OPENSTACK> cinder share show
/vx/fs1 *(rw,no_root_squash)
```

```
OPENSTACK> cinder share show
/vx/o_fs 2001:21::/120 (rw,sync,no_root_squash)
```

To share and export a file system

- ◆ To share and export a file system, enter the following:

```
OPENSTACK> cinder share add export-dir world|client
```

After issuing this command, OpenStack Cinder will be able to mount the exported file system using NFS.

export-dir Specifies the path of the directory that needs to be exported to the client.

The directory path should start with `/vx` and only the following characters are allowed:

`'a-zAZ0-9_/@+=.:-'`

world Specifies if the NFS export directory is intended for everyone.

client

Exports the directory with the specified options.

Clients may be specified in the following ways:

- **Single host**
Specify a host either by an abbreviated name recognized by the resolver, the fully qualified domain name, or an IP address.
- **Netgroups**
Netgroups may be given as `@group`. Only the host part of each netgroup member is considered when checking for membership.
- **IP networks**
You can simultaneously export directories to all hosts on an IP (sub-network). This is done by specifying an IP address and netmask pair as *address/netmask* where the netmask can be specified as a contiguous mask length. IPv4 or IPv6 addresses can be used.

To re-export new options to an existing share, the new options will be updated after the command is run.

For example:

```
OPENSTACK> cinder share add /vx/fs1 world
Exporting /vs/fs1 with options rw,no_root_squash

OPENSTACK> cinder share add /vx/o_fs 2001:21::/120
Exporting /vx/o_fs with options rw,sync,no_root_squash Success.
```

To delete the exported file system

- ◆ To delete (or unshare) the exported file system, enter the following:

```
OPENSTACK> cinder share delete export-dir client
```

For example:

```
OPENSTACK> cinder share delete /vx/fs1 world
Removing export path */vx/fs1
Success.
```

To start or display the status of the OpenStack Cinder service

- 1 To start the OpenStack Cinder service, enter the following:

```
OPENSTACK> cinder service start
```

The `OPENSTACK> cinder service start` command needs the NFS service to be up for exporting any mount point using NFS. The `OPENSTACK> cinder service start` command internally starts the NFS service by running the command `NFS> server start` if the NFS service has not been started. There is no `OPENSTACK> cinder service stop` command. If you need to stop NFS mounts from being exported, use the `NFS> server stop` command.

For example:

```
OPENSTACK> cinder server start
..Success.
```

- 2 To display the status of the OpenStack Cinder service, enter the following:

```
OPENSTACK> cinder service status
```

For example:

```
OPENSTACK> cinder server status
NFS Status on access_01 : ONLINE
NFS Status on access_02 : ONLINE
```

To display configuration changes that need to be done on the OpenStack controller node

- ◆ To display all the configuration changes that need to be done on the OpenStack controller node, enter the following:

```
OPENSTACK> cinder configure export-dir
```

export-dir

Specifies the path of the directory that needs to be exported to the client.

The directory path should start with `/vx` and only the following characters are allowed:

`'a-zAZ0-9_/@+.=:~'`

For example:

```
OPENSTACK> cinder configure /vx/fs1
```

To create a new volume backend named ACCESS_HDD in OpenStack Cinder

- 1 Add the following configuration block in the `/etc/cinder/cinder.conf` file on your OpenStack controller node.

```
enabled_backends=access-1
[access-1]
volume_driver=cinder.volume.drivers.veritas_cnfs.VeritasCNFSDriver
volume_backend_name=ACCESS_HDD
nfs_shares_config=/etc/cinder/access_share_hdd
nfs_mount_point_base=/cinder/cnfs/cnfs_sata_hdd
nfs_sparsed_volumes=True
nfs_disk_util=df
nfs_mount_options=nfsvers=3
```

Add the lines from the configuration block at the bottom of the file.

<code>volume_driver</code>	Name of the Veritas Access Cinder driver.
<code>volume_backend_name</code>	For this example, ACCESS_HDD is used. This name can be different for each NFS share. If several backends have the same name, the OpenStack Cinder scheduler decides in which backend to create the volume.
<code>nfs_shares_config</code>	This file has the share details in the form of <i>vip:/exported_dir</i> .
<code>nfs_mount_point_base</code>	Mount point where the share will be mounted on OpenStack Cinder. If the directory does not exist, create it. Make sure that the <code>Cinder</code> user has write permission on this directory.
<code>nfs_sparsed_volumes</code>	Preallocate or sparse files.
<code>nfs_disk_util</code>	Free space calculation.
<code>nfs_mount_options</code>	These are the mount options OpenStack Cinder uses to NFS mount.

This same configuration information for adding to the `/etc/cinder/cinder.conf` file can be obtained by running the `OPENSTACK CINDER> configure export_dir` command.

- 2 Append the following in the `/etc/cinder/access_share_hdd` file on your

OpenStack controller node:

```
vip:/vx/fs1
```

Use one of the virtual IPs for *vip*:

- 192.1.1.190
- 192.1.1.191
- 192.1.1.192
- 192.1.1.193
- 192.1.1.199

You can obtain Veritas Access virtual IPs using the `OPENSTACK> cinder configure export-dir` option.

- 3 Create the `/etc/cinder/access_share_hdd` file at the root prompt, and update it with the NFS share details.

```
# cnfs_sata_hdd(keystone_admin)]# cat /etc/cinder/access_share_hdd
192.1.1.190:/vx/fs1
```

- 4 The Veritas Access package includes the Veritas Access OpenStack Cinder driver, which is a Python script. The OpenStack Cinder driver is located at `/opt/VRTSnas/scripts/OpenStack/veritas_cnfs.py` on the Veritas Access node. Copy the `veritas_cnfs.py` file to `/usr/lib/python2.6/site-packages/cinder/volume/drivers/veritas_cnfs.py` if you are using the Python 2.6 release.

If you are using the OpenStack Kilo version of RDO, the file is located at:

```
/usr/lib/python2.7/site-packages/cinder/volume/drivers/veritas_cnfs.py
```

- 5 Make sure that the NFS mount point on the OpenStack controller node has the right permission for the cinder user. The cinder user should have write permission on the NFS mount point. Set the permission using the following command.

```
# setfacl -m u:cinder:rwX /cinder/cnfs/cnfs_sata_hdd

# sudo chmod -R 777 /cinder/cnfs/cnfs_sata_hdd
```

- 6 Give required permissions to the `/etc/cinder/access_share_hdd` file.

```
# sudo chmod -R 777 /etc/cinder/access_share_hdd
```

7 Restart the OpenStack Cinder driver.

```
# cnfs_sata_hdd(keystone_admin)]# /etc/init.d/openstack-cinder-volume
restart
Stopping openstack-cinder-volume: [ OK ]
Starting openstack-cinder-volume: [ OK ]
```

Restarting the OpenStack Cinder driver picks up the latest configuration file changes.

After restarting the OpenStack Cinder driver, `/vx/fs1` is NFS-mounted as per the instructions provided in the `/etc/cinder/access_share_hdd` file.

```
# cnfs_sata_hdd(keystone_admin)]# mount |grep /vx/fs1
192.1.1.190:/vx/fs1 on
cnfs_sata_hdd/e6c0baa5fb02d5c6f05f964423fecalf type nfs
(rw,nfsvers=3,addr=10.182.98.20)
```

You can obtain OpenStack Cinder log files by navigating to:

```
/var/log/cinder/volume.log
```

8 If you are using OpenStack RDO, use these steps to restart the OpenStack Cinder driver.

Login to the OpenStack controller node.

For example:

```
source /root/keystonerc_admin
```

Restart the services using the following command:

```
(keystone_admin)]# openstack-service restart openstack-cinder-volume
```

For more information, refer to the *OpenStack Administration Guide*.

9 On the OpenStack controller node, create a volume type named `va_vol_type`.

This volume type is used to link to the volume backend.

```
[root@c1059-r720xd-111046 cnfs_sata_hdd(keystone_admin)]#
cinder type-create va_vol_type
```

ID	Name
d854a6ad-63bd-42fa-8458-a1a4fadd04b7	va_vol_type

10 Link the volume type with the ACCESS_HDD back end.

```
[root@cl059-r720xd-111046cnfs_sata_hdd(keystone_admin)]# cinder type-key
va_vol_type set volume_backend_name=ACCESS_HDD
```

11 Create a volume of size 1gb.

```
[root@cl059-r720xd-111046 cnfs_sata_hdd(keystone_admin)]# cinder create --volume-type
va_vol_type --display-name va_vol1 1
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
created_at	2014-02-08T01:47:25.726803
display_description	None
display_name	va_vol1
id	disk ID 1
metadata	{}
size	1
snapshot_id	None
source_volid	None
status	creating
volume_type	va_vol_type

```
[root@cl059-r720xd-111046 cnfs_sata_hdd(keystone_admin)]# cinder list
```

ID	Status	Display Name	Size	Volume Type	Bootable	Attached to
disk ID 1	available	va_vol1	1	va_vol_type	false	

12 Extend the volume to 2gb.

```
[root@cl059-r720xd-111046 cnfs_sata_hdd(keystone_admin)]# cinder extend va_vol1 2
```

```
[root@cl059-r720xd-111046 cnfs_sata_hdd(keystone_admin)]# cinder list
```

ID	Status	Display Name	Size	Volume Type	Bootable	Attached to
disk ID 1	available	va_vol1	2	va_vol_type	false	

13 Create a snapshot.

```
[root@cl059-r720xd-111046 cnfs_sata_hdd(keystone_admin)]# cinder snapshot-create  
--display-name va_vol1-snap va_vol1
```

Property	Value
created_at	2014-02-08T01:51:17.362501
display_description	None
display_name	va_vol1-snap
id	disk ID 1
metadata	{}
size	2
status	creating
volume_id	52145a91-77e5-4a68-b5e0-df66353c0591

```
[root@cl059-r720xd-111046 cnfs_sata_hdd(keystone_admin)]# cinder snapshot-list
```

ID	Volume ID	Status	Display Name	Size
disk ID 1	52145a91-77e5-4a68-b5e0-df66353c0591	available	va_vol1-snap	2

14 Create a volume from a snapshot.

```
[root@cl059-r720xd-111046 cnfs_sata_hdd(keystone_admin)]# cinder  
create --snapshot-id e9dda50f-1075-407a-9cb1-3ab0697d274a --display-name  
va-vol2 2
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
created_at	2014-02-08T01:57:11.558339

About the Veritas Access integration with OpenStack Manila

OpenStack Cinder had the limitation of not being able to share a block device simultaneously between virtual machines. OpenStack Manila solves this problem.

OpenStack Manila provides a shared file system as a service. Using OpenStack Manila, you can share a single file system between multiple virtual machines.

Veritas Access is integrated with OpenStack Manila through a OpenStack Manila driver that lets you share Veritas Access file systems with virtual machines on OpenStack.

For the supported OpenStack versions for running the OpenStack Manila driver, see the *Veritas Access Installation Guide*.

The OpenStack Manila driver can create and manage simple file systems. For the backend to create simple file systems, use `va_fstype=simple` in the `manila.conf` file.

OpenStack Manila use cases

From the OpenStack controller node, an OpenStack administrator can do the following:

- Create and delete file systems.
- Allow and deny file system access to specific virtual machines.
- Provide IP-based access control.
- Create and delete snapshots of the file system.
- Provide free space statistics.
- NFS-based access of the shares from the instances.

Configuring Veritas Access with OpenStack Manila

To configure Veritas Access with OpenStack Manila

1 Export the pool to Manila.

```
OPENSTACK> manila resource export pool1
ACCESS Manila SUCCESS V-288-0 Pool exported to Manila
```

2 Enter the following command to configure the pool with Manila.

```
OPENSTACK> manila configure pool1
```

To create a new share backend va-share1 in Manila

Make the following changes on OpenStack controller node and restart the Manila driver. Add the following configuration entries in /etc/manila/manila.conf file:

In the [DEFAULT] section:

```
#####
enabled_share_backends=va-share1
#####
```

At the end of all sections:

```
#####
[va-share1]
share_driver=
manila.share.drivers.veritas.veritas_isa.ACCESSShareDriver
driver_handles_share_servers = False
share_backend_name = va-share1
va_server_ip = 10.209.106.144
va_port = 14161
va_fstype = simple
va_user = <master_user>
va_pwd = <master_user_password>
va_pool = pool1
#####
```

- 3 Enter the following command to display the resources which are created by Manila.

```
OPENSTACK> manila resource list
Pools exported to Manila: pool1
FS created by Manila:
FS snapshots created by Manila:
NFS shares exported by Manila:
```

Creating a new share backend on the OpenStack controller node

A backend is an instance of the OpenStack Manila share service, which is defined in a section of the `manila.conf` file. Each backend has exactly one driver.

To create a new share backend `va-share1` in OpenStack Manila, make the following changes on the OpenStack controller node, and restart the OpenStack Manila driver.

To create a new share backend on the OpenStack controller node

- 1 On the OpenStack controller node, add the following configuration entries in the OpenStack Manila `/etc/manila/manila.conf` file.
 - In the `DEFAULT` section, add the following:

```
#####
enabled_share_backends=va-share1
#####
```

If the entry `generic1` is already there, add the `va-share1` entry after a comma. For example:

```
enabled_share_backends = generic1,va-share1
```

- At the end of all sections in the `/etc/manila/manila.conf` file, add the following configuration entries:

```
#####
[va-share1]
share_driver= manila.share.drivers.veritas.veritas_isa.VeritasShareDriver
driver_handles_share_servers = False
share_backend_name = va-share1
va_server_ip = 10.182.96.179
va_port = 14161
va_fstype = simple
va_user = master
```

```
va_pwd = password
va_pool = pool1
####
```

The following table describes the options.

<code>share_backend_name</code>	Name of the share backend. This name can be different for each share backend.
<code>share_driver</code>	OpenStack Manila driver name.
<code>va_server_ip</code>	Console IP address of the Veritas Access cluster.
<code>va_port</code>	14161 The port on Veritas Access to which the Manila driver is connected.
<code>va_fstype</code>	Type of file system to be created on the specified pool. It can be <code>simple</code> .
<code>va_user</code>	Root user name.
<code>va_pwd</code>	Root password.
<code>va_pool</code>	Existing storage pool on Veritas Access from which the file systems are to be created.

You use the `OPENSTACK> manila configure` command to display the configuration options that need to be performed on the OpenStack controller node.

2 Restart the OpenStack Manila services.

The restart is on the OpenStack controller node, not on Veritas Access.

Creating an OpenStack Manila share type

An OpenStack Manila share type is an administrator-defined type of service that is used by the Manila scheduler to make scheduling decisions. OpenStack tenants can list share types and then use them to create new shares.

To create an OpenStack Manila share type

- ◆ On the OpenStack controller node, create a share type for `va-backend1` and `va-backend2`.

```
manila@C4110-R720xd-111045:~/OpenStack$ manila type-create va-backend1
False
```

To associate the share type to a share backend

- ◆ On the OpenStack controller node, associate the share type to a share backend.

```
manila@C4110-R720xd-111045:~/OpenStack$ manila type-key va-backend1 set  
driver_handles_share_servers=false share_backend_name=va-share1  
manila@C4110-R720xd-111045:~/OpenStack$ manila type-key va-backend2  
set driver_handles_share_servers=false share_backend_name=va-share2
```

Creating an OpenStack Manila file share

An OpenStack Manila file share is equivalent to a file system in Veritas Access.
You can create an OpenStack Manila file share on the OpenStack controller node.

To create an OpenStack Manila file share on the OpenStack controller node

- 1 On the OpenStack controller node, if you wanted to create two OpenStack Manila file shares called `prod_fs` and `finance_fs` of size 1 GB accessible over NFS, enter the following:

One of the file shares resides on `va_backend1`, and one of the file shares resides on `va-backend2`.

```
manila@C4110-R720xd-111045:~/OpenStack$ manila create --name prod_fs
--share-type va-backend1 NFS 1
```

```
manila@C4110-R720xd-111045:~/OpenStack$ manila create --name finance_fs
--share-type va-backend2 NFS 1
```

Use the `manila list` command to see how the file shares look on the OpenStack controller node.

You can see how the file systems look on Veritas Access as part of the share creation process.

SCSIini> storage fs list											
FS	STATUS	SIZE	LAYOUT	MIRRORS	COLUMNS	USE%	NFS SHARED	CIFS SHARED	FTP SHARED	SECONDARY TIER	
BE743021-F9FB406C	online	13.00G	simple	-	-	14%	yes	no	no	no	
D448A38C-E64392EC	online	1.00G	simple	-	-	5%	no	no	no	no	

2 Give `prod_fs` read-write access to 10.182.111.84.

```
manila@C4110-R720xd-111045:~/OpenStack$ manila access-allow --access-level rw  
ecba1f14-86b0-4460-a286-a7e938162fb4 ip 10.182.111.84
```

Property	Value
share_id	ecba1f14-86b0-4460-a286-a7e938162fb4
deleted	False
created_at	2015-04-28T17:59:45.514849
updated_at	None
access_type	ip
access_to	10.182.111.84
access_level	rw
state	new
deleted_at	None
id	8alc2d0b-a3fc-4405-a8eb-939adb8799db

In the `manila access-allow` command, you can get the ID (ecba1f14-86b0-4460-a286-a7e938162fb4) from the output of the `manila list` command.

3 Give `finance_fs` read-write access to 10.182.111.81.

```
manila@C4110-R720xd-111045:~/OpenStack$ manila access-allow --access-level rw  
f8da8ff6-15e6-4e0c-814b-d6ba8d08543c ip 10.182.111.81
```

Property	Value
share_id	f8da8ff6-15e6-4e0c-814b-d6ba8d08543c
deleted	False
created_at	2015-04-28T18:01:49.557300
updated_at	None
access_type	ip
access_to	10.182.111.81
access_level	rw
state	new
deleted_at	None
id	ddcfc2d2-7e71-443a-bd94-81ad05458e32

Use the `manila access-list <share-id>` command to display the different access given to instances.

Creating an OpenStack Manila share snapshot

You can create an OpenStack Manila share snapshot, which is equivalent to creating a snapshot (checkpoint) in Veritas Access. Creating an OpenStack Manila share snapshot creates a checkpoint of the specific file system on Veritas Access. The checkpoint that is created is non-removable.

Deleting a snapshot deletes the checkpoint of that file system.

To create an OpenStack Manila share snapshot

- ◆ On the OpenStack controller node, if you want to create `fin_snap` and `prod_snap` snapshots, enter the following:

```
manila@C4110-R720xd-111045:~/OpenStack$ manila snapshot-create --name fin_snap  
d3ab5cdc-4300-4f85-b4a5-e2a55d835031
```

```
manila@C4110-R720xd-111045:~/OpenStack$ manila snapshot-create --name prod_snap  
2269b813-0031-419e-a2d3-0073cdb2776e
```

Use the `manila snapshot-list` command to display the snapshots you created.

Managing Veritas Access storage services

- [Chapter 22. Deduplicating data](#)
- [Chapter 23. Compressing files](#)
- [Chapter 24. Configuring SmartTier](#)
- [Chapter 25. Configuring SmartIO](#)
- [Chapter 26. Configuring episodic replication](#)
- [Chapter 27. Configuring continuous replication](#)
- [Chapter 28. Using snapshots](#)
- [Chapter 29. Using instant rollbacks](#)
- [Chapter 30. Configuring Veritas Access with the NetBackup client](#)

Deduplicating data

This chapter includes the following topics:

- [About data deduplication](#)
- [Best practices for using the Veritas Access deduplication feature](#)
- [Setting up deduplication](#)
- [Configuring deduplication](#)
- [Manually running deduplication](#)
- [Scheduling deduplication](#)
- [Setting deduplication parameters](#)
- [Removing deduplication](#)
- [Verifying deduplication](#)

About data deduplication

Data deduplication is the process by which redundant data is eliminated to improve storage utilization. Using data deduplication, you can reduce the amount of storage required for storing user and application data. It is most effective in use-cases where many copies of very similar or even identical copies of data are stored. The deduplication feature in Veritas Access provides storage optimization for primary storage (storage of active data).

Each file in the configured file system is broken into user-configurable chunks for evaluating duplicates. The smaller the chunk size, the higher the percentage of sharing due to better chances of matches.

The first deduplication of a file system is always a full deduplication of the entire file system. This is an end-to-end deduplication process that identifies and eliminates

duplicate data. Any subsequent attempt to run deduplication on that file system results in incremental deduplication.

Note: Deduplication with a small chunk size increases the deduplication time and load on the system.

Veritas Access deduplication is periodic, that is, as per the user-configured frequency, redundant data in the file system is detected and eliminated.

Use cases for deduplication

The following are potential use cases for Veritas Access file system deduplication:

- Microsoft Exchange mailboxes
- File systems hosting user home directories
- Virtual Machine Disk Format (VMDK) or virtual image stores.

Relationship between physical and logical data on a file system

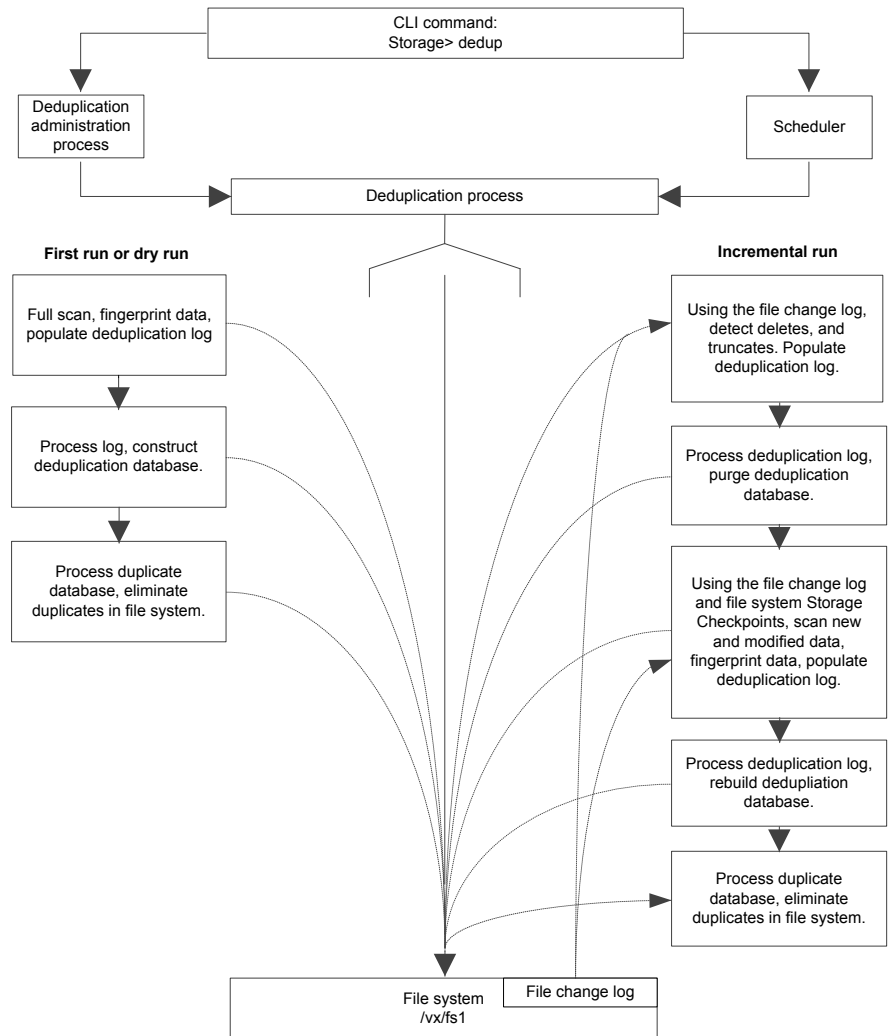
Table 22-1 shows an estimated file system data size that can be supported for a Veritas Access deduplicated file system.

Table 22-1 Relationship between physical and logical data on a file system for two billion unique fingerprints with various deduplication ratios

Fingerprint block size	Deduplication ratio	Unique signature per TB	Physical file system data size	Effective logical file system data size
4 K	50%	128 M	16 TB	32 TB
4 K	65%	90 M	23 TB	65 TB
4 K	80%	51 M	40 TB	200 TB
8 K	50%	64 M	32 TB	64 TB
8 K	65%	45 M	46 TB	132 TB
8 K	80%	25 M	80 TB	400 TB
16 K	50%	32 M	64 TB	128 TB
16 K	65%	22 M	93 TB	266 TB
16 K	80 %	13 M	158 TB	800 TB

Overview of the deduplication workflow

Figure 22-1 Overview of the deduplication workflow



The `Storage> dedup` commands perform administrative functions for the Veritas Access deduplication feature. The deduplication commands allow you to enable, disable, start, stop, and remove deduplication on a file system. You can also reset several deduplication configuration parameters and display the current deduplication status for your file system.

Note: Some configuration parameters can be set as local (specific to a file system) and or global (applicable to all deduplication-enabled file systems). Local parameters override the value of a global parameter.

Best practices for using the Veritas Access deduplication feature

The following are best practices when using the Veritas Access deduplication feature:

- Deduplication is most effective when the file system block size and the deduplication block size are the same for file systems with block sizes of 4K and above. This also allows the deduplication process to estimate space savings more accurately.
- The smaller the file system block size and the deduplication block size, the higher is the time required for performing deduplication. Smaller block sizes such as 1 KB and 2 KB, increase the number of data fingerprints that the deduplication database has to store.
Though the file system block size is data-dependent, the recommended block size for optimal deduplication is 4 KB for file systems less than 1 TB. For file systems 1 TB and above, it is 8 KB.
- For VMware NFS datastores that store Virtual Machine Disk Format (VMDK) images, a 4 KB block size is optimal.
- Compressed media files for images, music, and video, like JPEG, mp3, .MOV, and databases do not deduplicate or compress effectively.
- Home directory file systems are good candidates for deduplication.
- Deduplication is a CPU and I/O intensive process. It is a best practice to schedule deduplication when the load on your file systems is expected to be low.
- Evaluation of changes in the file system is done by the file system's File Change Log (FCL). Setting the frequency on a too infrequent basis may cause the FCL to rollover, thereby missing changes and deduplication opportunities to the file system.
- After enabling deduplication on file systems with existing data, the first deduplication run does a full deduplication. This can be time-consuming, and may take 12 to 15 hours per TB, so plan accordingly.
- The deduplication database takes up 1% to 7% of logical file system data. In addition, during deduplication processing, an additional but temporary storage space is required. Though 15% free space is enforced, it is recommended to

have 30% free space when the deduplication block size is less than 4096 (4 KB).

- If you plan to use the deduplication scheduler, you must have a Network Time Protocol (NTP) server enabled and configured.

Setting up deduplication

This is an end-to-end sample scenario of deduplication.

Note: This is not supported for a scale-out file system.

To deduplicate data

- 1 Ensure that the file system is deduplication-enabled. For example:

```
Storage> dedup list
```

See [“Configuring deduplication”](#) on page 327.

If the file system is not deduplication-enabled, you will need to enable it. For example:

```
Storage> dedup enable fs_name  
blksize
```

See [“Configuring deduplication”](#) on page 327.

- 2 (Optional) Once deduplication is enabled, you can set the CPU usage, the memory, and priority for the deduplication-enabled file system.

See [“Setting deduplication parameters”](#) on page 336.

- 3 Veritas recommends running a dryrun.

The dryrun provides you the space savings in percentage, if the threshold value is provided then the `fsdedup` command deduplicates data only if the expected savings cross the specified threshold.

```
Storage> dedup dryrun fs_name [threshold]
```

See [“Manually running deduplication”](#) on page 331.

Note: You cannot perform a dryrun on a file that has already been deduplicated.

- 4 You can either choose to start deduplication now or set up the deduplication schedule:

To start deduplication now

- Start deduplication now. For example:

```
Storage> dedup start fs_name [nodename]
```

See [“Manually running deduplication”](#) on page 331.

Note: If the system where you started deduplication crashes, the deduplication job fails over to one of the other nodes in the cluster. Run the `dedup status fs_name` command to find out the status. The `dedup status` command can temporarily show status as "FAILOVER" which means dedup job is currently being failed over and will resume shortly. dedup failover is applicable for deduplication jobs started with the `dedup start` command only. It is not applicable for scheduled dedup jobs.

To set up the deduplication schedule

- Set up the deduplication schedule. For example:

```
Storage> dedup schedule set fs_name hours day [freq]
```

See [“Scheduling deduplication”](#) on page 333.

- 5 You can check the status of the deduplication process. For example:

```
Storage> dedup status [fs_name]
```

See [“Verifying deduplication”](#) on page 339.

Configuring deduplication

To enable deduplication on a file system

- ◆ To enable deduplication on a file system, enter the following:

```
Storage> dedup enable fs_name blksize
```

Note: Deduplication must be enabled for a file system before setting file system configuration parameters and schedules.

This command also re-enables a deduplication schedule for a file system.

Enabling deduplication does not automatically deduplicate a file system.

Deduplication has to be manually started by using the `Storage> dedup start`

command or by setting up a schedule by using the `Storage> dedup set schedule` command.

fs_name Specify the file system name for which you want to enable deduplication.

blksize Specify the deduplication block size of the file system in bytes, where possible values of bytes are the following:

- `blksize=0` (Default)
- `blksize=1024`
- `blksize=2048`
- `blksize=4096`
- `blksize=8192`
- `blksize=16384`
- `blksize=32768`
- `blksize=65536`
- `blksize=131072`

Specify the deduplication block size in bytes, for example, 4096. The deduplication block size should be a power of 2. For example, 3 KB, is not a valid deduplication block size. The deduplication block size is a multiple of the file system's block size, and should be equal to or less than 128 KB.

0 is the default configuration for the deduplication block size.

If `blksize=0` is specified while enabling deduplication, then if the file system block size is < 4096, then the deduplication block size is set to 4096. Otherwise, the deduplication block size is set to the same as the file system block size.

Note: Once the deduplication block size is set when enabling file system deduplication, the deduplication block size cannot be changed. The only way to change the deduplication block size is to remove deduplication on the file system and then re-enable deduplication on the file system.

For example, to enable deduplication on the file system `fs1`, enter:

```
Storage> dedup enable fs1 blksize=4096
```

Note: For deduplication-enabled file systems, you are prompted to destroy the file system during `Storage> fs offline` and `Storage> fs destroy` operations.

To disable deduplication on a file system

- ◆ To disable deduplication on a file system, enter the following:

```
Storage> dedup disable fs_name
```

where *fs_name* is the name of the deduplication-enabled file system that you want to disable.

Only the deduplication schedule is suspended for a deduplication-disabled file system. All other configuration parameters, for example, file system configuration, schedule, and the deduplication database remain intact.

Note: Keeping a file system deduplication-disabled for a significant period of time may reduce the effectiveness of deduplication when it is re-enabled.

To list the deduplication-enabled file system or file systems

- ◆ To list the deduplication-enabled file system or file systems, enter the following:

```
Storage> dedup list fs_name
```

where *fs_name* is the name of the deduplication-enabled file system that you want to list.

For example, to list the deduplication-enabled file systems, *fs1*, and then *fs2*, enter:

```
Storage> dedup list fs1
```

```
Storage> dedup list fs2
```

Schedule hours are displayed as:

- * - is displayed as "Every hour"
- */N - is displayed as "Every N hours"
- 0,6,12,18,23 are shown as "00:00, 06:00, 12:00, 18:00, 23:00"

Note: 0 is replaced by 00:00, 1 is replaced by 01:00, 23 is replaced by 23:00

Schedule day interval is displayed as:

- * - is displayed as "Every day"
- */N - is displayed as "Every N days"
- 1 - is displayed as "Every Sunday"

- 2 - is displayed as "Every Monday"
- 3 - is displayed as "Every Tuesday"
- 4 - is displayed as "Every Wednesday"
- 5- is displayed as "Every Thursday"
- 6 - is displayed as "Every Friday"
- 7 - is displayed as "Every Saturday"

If you issue the command without *fs_name*, you get a list of all the deduplication-enabled file systems.

```
Storage> dedup list
```

The `Default` column header indicates the global value (applicable to all deduplication-enabled file systems). For example, if you have not set Priority, CPU, and Memory for file system *fs1*, the deduplication process uses the global value. Veritas Access deduplication uses the default values for global settings options. Local parameters override the value of global parameters.

Manually running deduplication

To create a deduplication dryrun

- ◆ To create a deduplication dryrun, enter the following command:

```
Storage> dedup dryrun fs_name [threshold]
```

The `Storage> dedup dryrun` command is useful for determining the statistics/potential savings on the file system data if actual deduplication is performed. Most accurate statistics are obtained when the file system block size and the deduplication block size are the same.

Note: You cannot perform a dryrun on a file system that has already been deduplicated.

<i>fs_name</i>	Specify the file system name for which you want to create a dryrun.
<i>threshold</i>	Specify the threshold percentage in the range of [0-100]. A dryrun is automatically converted to the actual deduplication if the dryrun meets the threshold value. For example, if you specified a threshold value of 40%, and if deduplication results in a space savings of $\geq 40\%$, then the dryrun is automatically converted to the actual deduplication

To check whether the deduplication dryrun reaches to a threshold value of 60%, enter the following:

```
Storage> dedup dryrun fs1 60
```

To start the deduplication process

- ◆ To manually start the deduplication process, enter the following:

```
Storage> dedup start fs_name [nodename]
```

where *fs_name* is the name of the file system where you want to start the deduplication process and *nodename* is the node in the cluster where you want to start deduplication. You can run deduplication on any node in the cluster.

Note: If the system where you started deduplication crashes, the deduplication job fails over to one of the other nodes in the cluster. Run the `dedup status fs_name` command to find out the status. The `dedup status` command can temporarily show status as "FAILOVER" which means dedup job is currently being failed over and will resume shortly. dedup failover is applicable for deduplication jobs started with the `dedup start` command only. It is not applicable for scheduled dedup jobs.

When the deduplication process is started for the first time, a full scan of the file system is performed. Any subsequent attempt to run deduplication requires an incremental scan only.

For example:

```
Storage> dedup start fs1 node_01
```

Note: When deduplication is running on a file system, you run the `Storage> fs offline` or `Storage> fs destroy` commands, these operations can proceed only after deduplication is stopped by using the `Storage> dedup stop` command.

To stop the deduplication process

- ◆ To stop the deduplication process running on a file system, enter the following command:

```
Storage> dedup stop fs_name
```

where *fs_name* is the name of the file system where you want to stop the deduplication process.

Note: The deduplication process may not stop immediately as a consistent state is ensured while stopping. Use the `Storage> dedup status` command to verify if the deduplication process has stopped.

Scheduling deduplication

To set the deduplication schedule

- 1 To set the deduplication schedule, enter the following:

```
Storage> dedup schedule set fs_name  
      hours  
      day [freq]
```

The `Storage> dedup schedule set` command can only be set as a local parameter.

Two categories of schedules are allowed: run periodicity and type periodicity. The granularity of the schedule is limited to the time of day and the day of the month.

<i>fs_name</i>	Specify the file system where you want to set the deduplication schedule.
----------------	---------------------------------------------------------------------------

<i>hours</i>	Specify the <code>hours</code> value for setting the duplication schedule.
--------------	----------------------------------------------------------------------------

There are three types of values for the `hours` field:

- * - indicates every hour.
- */N - indicates every Nth hour, where N is in the range [1-12].
- You can also specify 5 comma-separated hours in the range of [0-23].

day Specify the interval in days for setting the deduplication schedule.

There are three types of values for the `day` field:

- `*` - indicates every day.
- `*/N` - indicates every Nth day, where N is in the range of [1-15].
- Any number in the range of [1-7] where:
 - 1 - Sunday
 - 2 - Monday
 - 3 - Tuesday
 - 4 - Wednesday
 - 5 - Thursday
 - 6 - Friday
 - 7 - Saturday

The deduplication scheduler will only pick up the schedule if the schedule is enabled for deduplication.

freq Specify the frequency to run the deduplication schedule in the range of [1-5]. The default frequency is [1].

This value controls deduplication load on the file system by distributing phases of deduplication across various runs, and potentially across systems in the cluster. A value of 4 means, every 4th run deduplicates the file system, whereas the other runs consolidate the changes.

2 When you set a deduplication schedule, keep in mind the following:

- If the hour value for a schedule is set as `*/N`, then the deduplication scheduler picks up the schedule every Nth hour starting from 00:00 and ending at 23:00. The hour schedule resets at end of day.
For example, if the hour value is `*/5`, then the schedule time will be 00:00, 05:00, 10:00, 15:00, and 20:00 hours. On the next day, the schedule repeats at the same times.
- If day value for a schedule is set as `*/N`, then the deduplication scheduler picks up the schedule every Nth day starting from the 1st day of the month and ending with the last day of the month. The day schedule resets at end of each month.
For example, if the day value is `*/5`, then the schedule day is on the 1st, 6th, 11th, 16th, 21st, 26th, and 31st days for a 31-day month. For the next month, the schedule repeats on the same days.
- For both the hour and day schedule, the `*` and `*/1` values are interchangeable. They indicate every hour and every day.

To modify the deduplication schedule

- ◆ To modify the deduplication schedule, enter the following:

```
Storage> dedup schedule modify fs_name  
             hours  
             day  
             freq
```

fs_name Specify the file system where you want to modify the deduplication schedule.

hours Specify the `hours` value for modifying the deduplication schedule.
There are three types of values for the `hours` field:

- * - indicates every hour.
- */N - indicates every Nth hour, where N is in the range [1-12].
- You can also specify 5 comma-separated hours in the range of [0-23].

day Specify the interval in days for modifying the deduplication schedule.

There are three types of values for the `day` field:

- * - indicates every day.
- */N - indicates every Nth hour, where N is in the range [1-15].
- Any number in the range of [1-7] where:
 - 1 - Sunday
 - 2 - Monday
 - 3 - Tuesday
 - 4 - Wednesday
 - 5 - Thursday
 - 6 - Friday
 - 7 - Saturday

Note: The deduplication scheduler will only pick up the schedule if the schedule is enabled for deduplication.

freq Specify the frequency to run the deduplication schedule in the range of [1-5].

To delete the deduplication schedule

- ◆ To delete the deduplication schedule, enter the following:

```
Storage> dedup schedule delete fs_name
```

where *fs_name* is the name of the file system that you want to delete.

Setting deduplication parameters

To set the CPU usage for the deduplication-enabled file system

- ◆ To set the CPU usage for the file system, enter the following:

```
Storage> dedup set cpu cpuvalue fs_name
```

cpuvalue

Specify the CPU usage behavior for the deduplication-enabled file system.

The following are the available values:

- IDLE - indicates that the deduplication process consumes as much CPU processing as is available. For example, if the CPUs are IDLE, then the deduplication process takes all of the idle CPUs, and performs the deduplication job faster. CPU usage may reach 100% on each available CPU.
- YIELD (default) - indicates that the deduplication process yields the CPU periodically; that is, even if the CPUs are not busy, the deduplication process relinquishes the CPU. More time may be taken for the same job in some scenarios. However, the yield value ensures that the deduplication process does not hang onto the CPU, or cause CPU usage spikes.

fs_name

Specify the deduplication-enabled file system for which you want to set the CPU usage.

Note: If a file system name is specified, the `Storage> dedup set cpu` command sets the CPU value for that file system. Otherwise, the CPU value is applicable to all file systems, which have not overridden the CPU value.

To set the deduplication memory allocation limit for the deduplication-enabled file system

- ◆ To set the deduplication memory limit in MB for the deduplication-enabled file system, enter the following:

```
Storage> dedup set memory memvalue
```

where *memvalue* is the memory value in MB, for example, 1024.

The *memvalue* controls the maximum memory per deduplication process.

Note: Care must be taken to increase *memvalue* if large file systems are present. Otherwise, deduplication efficiency may be affected. Since this is a limit value, only the required memory is consumed for smaller file system deduplication jobs. Note that scheduled deduplication jobs start deduplication based on the available memory; therefore, if available RAM in the system falls below the configured memory allocation limit for deduplication, the deduplication scheduler on that system postpones the scheduled deduplication. At this point, other systems with available memory starts deduplication. If the job remains postponed for 1 hour, the job will be abandoned.

To set the deduplication priority for the deduplication-enabled file system

- ◆ To set the deduplication priority (importance) for the deduplication-enabled file system, enter the following:

```
Storage> dedup set priority priorityvalue [fs_name]
```

priorityvalue

Specify the importance of deduplication for the file system. The setting of this parameter is local (specific to a file system). The *priorityvalue* parameter is used by the deduplication scheduler to evaluate if starting deduplication at the scheduled time is appropriate or not based on the state of the file system at that time.

priorityvalue is also a load-balancing mechanism whereby a less-loaded system in the cluster may pick up a scheduled deduplication job.

Available values are the following:

- LOW (default) - indicates that if the system has sustained CPU usage of 50% or more in the last one hour, the file systems marked as LOW have their deduplication schedules skipped with a message in the syslog
- NORMAL - indicates that if the system has sustained CPU usage of 80% or more in the last one hour, the file systems marked as NORMAL have their deduplication schedules skipped with a message in the syslog
- HIGH - indicates that starting deduplication is a must for this file system, and without evaluating any system state, deduplication is started at the scheduled time.

fs_name

Specify the file system where you want to set the deduplication priority.

Removing deduplication

To remove deduplication configuration-related information from the specified file system

- ◆ Enter the following:

```
Storage> dedup remove fs_name
```

where *fs_name* is the name of the file system for which you want to remove deduplication.

This command removes all configurations and the deduplication database for the specified file system.

Note: This operation cannot be undone, and re-running deduplication on your file system may take a significant amount of time.

Verifying deduplication

To obtain status information for a specified deduplicated-enabled file system or all deduplicated-enabled file systems

- ◆ Enter the following command:

```
Storage> dedup status [fs_name]
```

where *fs_name* is the specified deduplicated-enabled file system for which you want to obtain current status information.

If you issue the command without *fs_name*, you get status information for all deduplicated-enabled file systems. For example:

If you issue the command with *fs_name*, you get the detailed status information for the specified file system, along with any error messages or warnings.

The following describes the output from the `Storage> dedup status` command:

Filesystem	Displays the directory where the file system is mounted.
Savings	<p>Displays the savings as a percentage. The value can mean different things during the course of deduplication.</p> <p>When the deduplication is in a COMPLETED state, or when the deduplication process is computing the expected deduplication, the value in this column shows the actual sharing in the file system. However, when the expected deduplication calculation is complete, this column value shows the expected deduplication. The expected deduplication calculation is based on user data only; therefore, at the end of deduplication, the saving percentage may vary from the expected deduplication percentage. This is because the actual file system deduplication percentage not only takes into consideration the user data, but also file system and deduplication metadata. This difference may be pronounced if the user data is very small. For a failed deduplication, the value is undefined.</p>
Status	<p>Displays one of the following status values:</p> <ul style="list-style-type: none">■ RUNNING■ COMPLETED■ STOPPED■ FAILED■ FAILOVER■ NONE - indicates that deduplication has not been previously run on this file system.

Node	Indicates the node name where the deduplication job is either running or has completed for a file system.
Type	<p>The following are the types of deduplication jobs:</p> <ul style="list-style-type: none"> ■ MANUAL - the deduplication job is started by using either the <code>Storage> dedup start</code> command or the <code>Storage> dedup dryrun</code> command. ■ SCHEDULED - the deduplication job is started by the deduplication scheduler.
Details	<p>Displays the status of the file system deduplication activity.</p> <p>The deduplication process writes its status in the status log. The relevant status log is displayed in this column. For a long-running deduplication process, the status log may also show the actual file system sharing as a progress indicator. This actual file system sharing percentage along with the expected saving percentage in the <code>Saving</code> column gives a good estimate of the progress. When displaying deduplication status for a specific file system, any errors, or warnings for the deduplication run are also shown. The <code>Details</code> column gives a detailed idea of what to look for in case of any issues.</p>

Compressing files

This chapter includes the following topics:

- [About compressing files](#)
- [Use cases for compressing files](#)
- [Best practices for using compression](#)
- [Compression tasks](#)

About compressing files

Compressing files reduces the space used, while retaining the accessibility of the files and being transparent to applications. Compressed files look and behave almost exactly like uncompressed files: the compressed files have the same name, and can be read and written as with uncompressed files. Reads cause data to be uncompressed in memory, only; the on-disk copy of the file remains compressed. In contrast, after a write, the new data is uncompressed on disk.

Only user data is compressible. You cannot compress Veritas File System (VxFS) metadata.

After you compress a file, the inode number does not change, and file descriptors opened before the compressions are still valid after the compression.

Compression is a property of a file. Thus, if you compress all files in a directory, for example, any files that you later copy into that directory do not automatically get compressed. You can compress the new files at any time by compressing the files in the directory again.

You compress files with the `Storage> compress` command.

See [“Compression tasks”](#) on page 343.

See the `storage_compress(1)` manual page.

To compress files, you must have VxFS file systems with disk layout Version 8 or later.

See [“Upgrading disk layout versions”](#) on page 248.

Note: When you back up compressed files to tape, the backup program stores the data in an uncompressed format. The files are uncompressed in memory and subsequently written to the tape. This results in increased CPU and memory usage when you back up compressed files.

About the compressed file format

A compressed file is a file with compressed extents. A `compress` call compresses all extents of a file. However, writes to the file cause the affected extents to get uncompressed; the result can be files with both compressed and uncompressed extents.

About the file compression attributes

When you compress a file with the `Storage> compress` command, `compress` attaches the following information to the inode:

- Compression algorithm
- Compression strength, which is a number from 1 to 9
- Compression block size

This information is referred to as the file compression attributes. The purpose of the attributes are to collect the parameters used to create the compressed file. The information can then be read by a backup program.

The file compression attributes guarantee that a particular compressed file can only use one type and strength of compression. Recompressing a file using different attributes fails. To change the file compression attributes, you must explicitly uncompress first, and then recompress with the new options, even in the case where all extents are already uncompressed.

The file compression attributes do not indicate if all extents are compressed. Some extents might be incompressible, and other extents or even all extents might be uncompressed due to writes, but the file compression attributes remain. Only an explicit file uncompression can remove the attributes.

About the file compression block size

The file compression algorithm compresses data in the specified block size, which defaults to 1MB. Each compression block has its own extent descriptor in the inode. If the file or the last extent is smaller than the compression block size, then that smaller size gets compressed. The maximum block size is 1MB.

Extents with data that cannot be compressed are still marked as compressed extents. Even though such extents cannot be compressed, marking these extents as compressed allows successive compression runs to skip these extents to save time. Shared extents cannot be compressed and do not get marked as compressed. Since the file compression algorithm looks at fixed-size blocks, the algorithm finds these incompressible extents in units of the file compression block size.

Use cases for compressing files

The following list contains common use case categories:

- If files are old and not accessed frequently. For example:
 - Compress database archive logs which are older than 8 days.
 - Compress jpeg files which are not accessed in 30 days.

Best practices for using compression

Best practices for using compression:

- Schedule compression during non-peak hours.

Compression tasks

Table 23-1 Compression tasks

How to	Task
How to compress a file or all files in a directory	See “Compressing files” on page 344.
How to scheduled compression jobs	See “Scheduling compression jobs” on page 345.
How to list compressed files	See “Listing compressed files” on page 346.
How to show the scheduled compression job	See “Scheduling compression jobs” on page 345.

Table 23-1 Compression tasks (*continued*)

How to	Task
How to uncompress a file or all files in a directory	See “Uncompressing files” on page 346.
How to modify the scheduled compression	See “Modifying the scheduled compression” on page 347.
How to remove the specified schdule.	See “Removing the specified schedule” on page 348.
How to stop the schedule for a file system.	See “Stopping the schedule for a file system” on page 349.
How to remove the pattern-related rule for a file system	See “Removing the pattern-related rule for a file system” on page 349.
How to remove the modification age (age-based) related rule for a file system	See “Removing the modified age related rule for a file system” on page 349.

Compressing files

You can compress a file or compress all files in a directory.

Note: This is not supported for a scale-out file system.

To compress a file

- ◆ Compress a file:

```
Storage> compress file fs_name file_or_dir resource_level algorithm
```

where *fs_name* is the name of the file system.

where *file_or_dir* is the name of the file or directory.

where *resource_level* is either low, medium, or high.

where *algorithm* is the file compression algorithm strength [1–9]. For example, you specify strength gzip-3 compression as "3".

See [“About the file compression attributes”](#) on page 342.

To compress all files in a directory

- ◆ Compress all files in a directory:

```
Storage> compress file fs_name file_or_dir resource_level algorithm
```


Showing the scheduled compression job

To show the scheduled compression job

- ◆ Show the scheduled compression job

```
Storage> compress schedule show new_schedule
```

where *new_schedule* is the name of the schedule.

Scheduling compression jobs

Schedule compression jobs lets you compress pattern-based and age-based compression.

To schedule compression

- 1 Create a scheduled compression:

```
Storage> compress schedule create new_schedule duration min \  
[hour] [day_of_month] [month] [day_of_week] [node]
```

where *new_schedule* is the name of the schedule.

where *duration* is the duration specified in hours (1 or more).

where *min* is the minutes.

where *hour* is the hours.

where *day* is the day of the month.

where *month* is the month.

where *day_of_week* is the day of the week.

where *node* is the name of the node or you can use "any".

- 2 Start the schedule for a given file system:

```
Storage> compress schedule start fs_name schedule_name \  
resource_level algorithm
```

where *fs_name* is the name of the file system.

where *schedule_name* is the name of the schedule.

where *resource_level* is either `low`, `medium`, or `high`.

where *algorithm* is the file compression algorithm strength [1-9]. For example, you specify strength gzip-3 compression as "3".

3 Show the scheduled compression:

```
Storage> compress schedule show new_schedule
```

4 (Optional) Create a pattern for the file system.

```
Storage> compress pattern create fs_name pattern
```

where *pattern* is the extensions of the file names separated by ", " For example, *.arc, *.dbf, *.tmp.

5 (Optional) Create a modification age rule for the file system.

```
Storage> compress modage create fs_name mod_age
```

where *mod_age* is the modification age (age-based) specified units are in days.

6 If you performed step 4 or 5, you can list the schedule details for the file system:

```
Storage> compress schedule list fs_name
```

Listing compressed files

To list compressed files

◆ List compressed files:

```
Storage> compress list fs_name file_or_dir
```

where *fs_name* is the name of the file system.

where *file_or_dir* is the name of the file or directory.

Uncompressing files

To uncompress a file

◆ Uncompress a file:

```
Storage> uncompress file fs_name file_or_dir resource_level
```

where *fs_name* is the name of the file system.

where *file_or_dir* is the name of the file or directory.

where *resource_level* is either low, medium, or high.

To uncompress all files in a directory

- ◆ Uncompress all files in a directory:

```
Storage> uncompress file fs_name file_or_dir resource_level
```

Modifying the scheduled compression

To change the scheduled compression

- 1 Stops the schedule for the file system:

```
Storage> compress schedule stop fs_name
```

where *fs_name* is the name of the file system.

- 2 Remove specified schedule:

```
Storage> compress schedule remove new_schedule
```

3 Create a scheduled compression:

```
Storage> compress schedule create new_schedule duration min \  
[hour] [day_of_month] [month] [day_of_week] [node]
```

where *new_schedule* is the name of the schedule.

where *duration* is the duration specified in hours (1 or more).

where *min* is the minutes.

where *hour* is the hours.

where *day* is the day of the month.

where *month* is the month.

where *day_of_week* is the day of the week.

where *node* is the name of the node or you can use "any".

4 Start the schedule for a given file system:

```
Storage> compress schedule start fs_name schedule_name \  
resource_level algorithm
```

where *fs_name* is the name of the file system.

where *schedule_name* is the name of the schedule.

where *resource_level* is either `low`, `medium`, or `high`.

where *algorithm* is the file compression algorithm strength [1-9]. For example, you specify strength gzip-3 compression as "3".

Removing the specified schedule

To remove the specified schedule

- ◆ Enter the following:

```
Storage> compress schedule remove new_schedule
```

where *new_schedule* is the name of the schedule.

Stopping the schedule for a file system

To stop the schedule for a file system

- ◆ Enter the following:

```
Storage> compress schedule stop fs_name
```

where *fs_name* is the name of the file system.

Removing the pattern-related rule for a file system

To remove the pattern-related rule a named file system

- ◆ Enter the following:

```
Storage> compress pattern remove fs_name
```

where *fs_name* is the name of the file system.

Removing the modified age related rule for a file system

To remove the modified age related rule for a file system

- ◆ Enter the following:

```
Storage> compress modage remove fs_name
```

where *fs_name* is the name of the file system.

Configuring SmartTier

This chapter includes the following topics:

- [About Veritas Access SmartTier](#)
- [How Veritas Access uses SmartTier](#)
- [Configuring the policy of each tiered file system](#)
- [Adding tiers to a file system](#)
- [Adding or removing a column from a secondary tier of a file system](#)
- [Configuring a mirror to a tier of a file system](#)
- [Listing all of the files on the specified tier](#)
- [Displaying a list of SmartTier file systems](#)
- [About tiering policies](#)
- [About configuring the policy of each tiered file system](#)
- [Best practices for setting relocation policies](#)
- [Relocating a file or directory of a tiered file system](#)
- [Displaying the tier location of a specified file](#)
- [About configuring schedules for all tiered file systems](#)
- [Configuring schedules for tiered file systems](#)
- [Displaying the files that may be moved or pruned by running a policy](#)
- [Allowing metadata information on the file system to be written on the secondary tier](#)
- [Restricting metadata information to the primary tier only](#)

- [Removing a tier from a file system](#)

About Veritas Access SmartTier

The Veritas Access SmartTier feature makes it possible to allocate two tiers of storage to a file system.

The following features are part of the Veritas Access SmartTier solution:

- Relocate files between primary and secondary tiers automatically as files age and become less business critical.
- Prune files on secondary tiers automatically as files age and are no longer needed.
- Promote files from a secondary storage tier to a primary storage tier based on I/O temperature.
- Retain original file access paths to eliminate operational disruption, for applications, backup procedures, and other custom scripts.
- Let you manually move folders, files and other data between storage tiers.
- Enforce the policies that automatically scan the file system and relocate files that match the appropriate tiering policy.

Note: The SmartTier functionality is available only for a Cluster File System and not for a scale-out file system.

In Veritas Access, there are two predefined tiers for storage:

- Current active tier 1 (primary) storage.
- Tier 2 (secondary) storage for aged or older data.

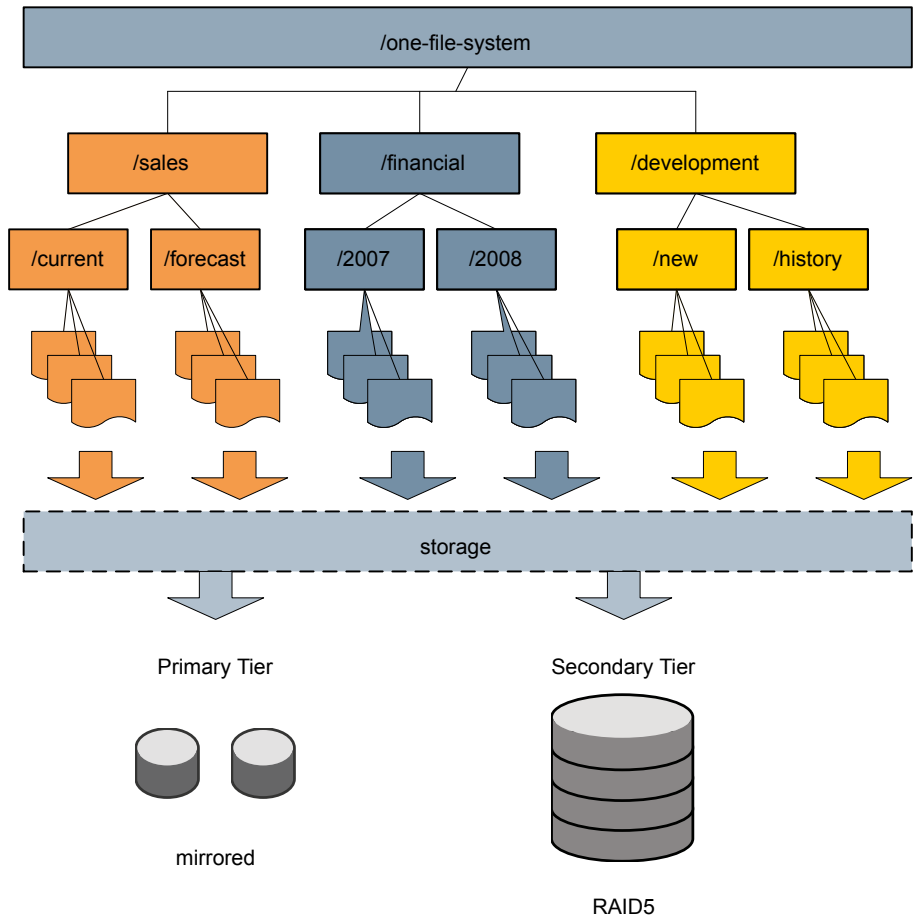
To configure Veritas Access SmartTier, add tier 2 (secondary) storage to the configuration. Specify where the archival storage resides (storage pool) and the total size.

Files can be moved from the active storage after they have aged for a specified number of days, depending on the policy selected. The number of days for files to age (not accessed) before relocation can be changed at any time.

Note: An aged file is a file that exists without being accessed.

[Figure 24-1](#) depicts the features of Veritas Access and how it maintains application transparency.

Figure 24-1 SmartTier features



If you are familiar with Veritas Volume Manager (VxVM), every Veritas Access file system is a multi-volume file system (one file system resides on two volumes). The SmartTier tiers are predefined to simplify the interface. When an administrator wants to add storage tiering, a second volume is added to the volume set, and the existing file system is encapsulated around all of the volumes in the file system.

How Veritas Access uses SmartTier

Veritas Access provides two types of tiers:

- Primary tier

- Secondary tier

Each newly created file system has only one primary tier initially. This tier cannot be removed.

For example, the following operations are applied to the primary tier:

```
Storage> fs addmirror
```

```
Storage> fs growto
```

```
Storage> fs shrinkto
```

The `Storage> tier` commands manage file system tiers.

All `Storage> tier` commands take a file system name as an argument and perform operations on the combined construct of that file system.

The Veritas Access file system default is to have a single storage tier. An additional storage tier can be added to enable storage tiering. A file system can only support a maximum of two storage tiers.

`Storage> tier` commands can be used to perform the following:

- Adding/removing/modifying the secondary tier
- Setting policies
- Scheduling policies
- Locating tier locations of files
- Listing the files that are located on the primary or the secondary tier
- Moving files from the secondary tier to the primary tier
- Allowing metadata information on the file system to be written on the secondary tier
- Restricting metadata information to the primary tier only

Configuring the policy of each tiered file system

To display the policy of each tiered file system

- ◆ To display the policy of each tiered file system, enter the following:

Storage> tier policy list

For example:

```
Storage> tier policy list
FS      Create on  Days  MinAccess Temp  PERIOD
==      =====  ====  =====
fs1     primary    2      3              4
```

Each tiered file system can be assigned a policy. A policy that is assigned to a file system has three parts:

file creation	Specifies on which tier the new files are created.
inactive files	Indicates when a file has to be moved from the primary tier to the secondary tier. For example, if the <code>days</code> option of the tier is set to 10, and if a file has not been accessed for more than 10 days, then it is moved from the primary tier of the file system to the secondary tier.
access temperature	Measures the number of I/O requests to the file during the period as designated by the period. In other words, it is the number of read or write requests that are made to a file over a specified number of 24-hour periods, divided by the number of periods. If the access temperature of a file exceeds <code>minacctemp</code> (where the access temperature is calculated over a period of time previously specified) then this file is moved from the secondary tier to the primary tier.

To modify the policy of a tiered file system

- ◆ To modify the policy of a tiered file system, enter the following:

```
Storage> tier policy modify fs_name {primary|secondary} days
minacctemp period
```

fs_name	The name of the tiered file system from which you want to modify a policy.
tier	Causes the new files to be created on the primary or the secondary tier. You need to input either <code>primary</code> or <code>secondary</code> .
days	Number of days from which the inactive files move from the primary to the secondary tier.
minacctemp	The minimum access temperature value for moving files from the secondary to the primary tier.
period	The number of past days used for calculating the access temperature.

To display the prune policy of a tiered file system

- ◆ To display the prune policy of a tiered file system, enter the following:

```
Storage> tier policy prune list
```

By default, the prune policy status of a tiered file system is `disabled`. The `delete_after` indicates the number of days after which the files can be deleted.

To modify the prune policy of a tiered file system

- ◆ To modify the prune policy of a tiered file system, enter the following:

```
Storage> tier policy prune modify fs_name delete_after
```

fs_name	Name of the tiered file system from which you want to modify the prune policy.
delete_after	Number of days after which the inactive files are deleted.

To remove the prune policy of a tiered file system

- ◆ To remove the prune policy of a tiered file system, enter the following:

```
Storage> tier policy prune remove fs_name
```

where *fs_name* is the name of the tiered file system from which you want to remove the prune policy.

To run the policy of a tiered file system

- ◆ To run the policy of a tiered file system, enter the following:

```
Storage> tier policy run fs_name
```

where *fs_name* indicates the name of the tiered file system for which you want to run a policy.

To remove the policy of a tiered file system

- ◆ To remove the policy of a tiered file system, enter the following:

```
Storage> tier policy remove fs_name
```

where *fs_name* indicates the name of the tiered file system from which you want to remove a policy.

You can run the policy of a tiered file system, which would be similar to scheduling a job to run your policies, except in this case running the policy is initiated manually. The `Storage> tier policy run` command moves the older files from the primary tier to the secondary tier, or prunes the inactive files on the secondary tier, according to the policy setting.

Adding tiers to a file system

You can add the following types of tiers to file systems:

- simple
- mirrored
- striped
- mirrored-stripe
- striped-mirror

Note: If a rollback exists for the file system, adding a tier can cause inconsistencies in the rollback hierarchy. The recommended method is to create the tier first and then create the rollback.

To add a second tier to a file system

- ◆ To add a tier to a file system where the layout is "simple" (concatenated), enter the following:

```
Storage> tier add simple fs_name size pool1[,disk1,...]
```

To add a mirrored tier to a file system

- ◆ To add a mirrored tier to a file system, enter the following:

```
Storage> tier add mirrored fs_name size nmirrors  
pool1[,disk1,...] [protection=disk|pool]
```

To add a striped tier to a file system

- ◆ To add a striped tier to a file system, enter the following:

```
Storage> tier add striped fs_name size ncolumns  
pool1[,disk1,...] [stripeunit=kilobytes]
```

To add a mirrored-striped tier to a file system

- ◆ To add a mirrored-striped tier to a file system, enter the following:

```
Storage> tier add mirrored-stripe fs_name size nmirrors ncolumns  
pool1[,disk1,...] [protection=disk|pool] [stripeunit=kilobytes]
```

To add a striped-mirror tier to a file system

- ◆ To add a striped-mirror tier to a file system, enter the following:

```
Storage> tier add striped-mirror fs_name size nmirrors ncolumns  
pool1[,disk1,...] [protection=disk|pool] [stripeunit=kilobytes]
```

fs_name	Specifies the name of the file system to which the mirrored tier is added. If the specified file system does not exist, an error message is displayed.
size	Specifies the size of the tier to be added to the file system (for example, 10m, 10M, 25g, 100G).
ncolumns	Specifies the numbers of columns to add to the striped tiered file system.
nmirrors	Specifies the number of mirrors to be added to the tier for the specified file system.

pool1[,disk1,...]	<p>Specifies the pool(s) or disk(s) that is used for the specified tiered file system. If the specified pool or disk does not exist, an error message is displayed. You can specify more than one pool or disk by separating the pool or the disk name with a comma, but do not include a space between the comma and the name.</p> <p>The disk needs to be part of the pool or an error message is displayed.</p>
protection	<p>If no protection level is specified, <code>disk</code> is the default protection level.</p> <p>The protection level of the second tier is independent of the protection level of the first tier.</p> <p>Available options are:</p> <ul style="list-style-type: none"> ■ <code>disk</code> - If <code>disk</code> is entered for the protection field, then mirrors are created on separate disks. The disks may or may not be in the same pool. ■ <code>pool</code> - If <code>pool</code> is entered for the protection field, then mirrors are created in separate pools. If not enough space is available, then the file system is not created.
stripeunit=kilobytes	<p>Specifies a stripe width of either 128K, 256k, 512K, 1M, or 2M. The default stripe width is 512K.</p>

Adding or removing a column from a secondary tier of a file system

You can add a column to a secondary tier of a file system.

To add a column to a secondary tier of a file system

- ◆ To add a column to a secondary tier of a file system, enter the following:

```
Storage> tier addcolumn fs_name ncolumns pool_or_disk_name
```

fs_name Specifies the file system for which you want to add a column to a secondary tier of the file system.

ncolumn Specifies the number of columns that you want to add to the secondary tier of the file system.

Note: In the case of striped file systems, the number of disks that are specified should be equal to the number of columns (*ncolumns*).

Note: In the case of mirrored-striped and striped-mirrored file systems, the disks that are specified should be equal to (*ncolumns * number_of_mirrors_in_fs*).

pool_or_disk_name Specifies the pool or the disk name for the tiered file system.

To remove a column from a secondary tier of a file system

- ◆ To remove a column from a secondary tier of a file system, enter the following:

```
Storage> tier rmcolumn fs_name
```

where *fs_name* is the name of the tiered file system, the secondary tier of which you want to remove the column from.

Configuring a mirror to a tier of a file system

To add a mirror to a tier of a file system

- ◆ To add a mirror to a tier of a file system, enter the following:

```
Storage> tier addmirror fs_name pool1[,disk1,...]  
[protection=disk|pool]
```

fs_name Specifies the file system to which a mirror is added. If the specified file system does not exist, an error message is displayed.

<code>pool1[,disk1,...]</code>	<p>Specifies the pool(s) or disk(s) that are used as a mirror for the specified tiered file system. You can specify more than one pool or disk by separating the name with a comma. But do not include a space between the comma and the name.</p> <p>The disk needs to be part of the pool or an error message is displayed.</p>
<code>protection</code>	<p>If no protection level is specified, <code>disk</code> is the default protection level.</p> <p>Available options are:</p> <ul style="list-style-type: none"> ■ <code>disk</code> - If <code>disk</code> is entered for the protection field, then mirrors are created on separate disks. The disks may or may not be in the same pool. ■ <code>pool</code> - If <code>pool</code> is entered for the protection field, then mirrors are created in separate pools. If not enough space is available, then the file system is not created.

To remove a mirror from a tier of a file system

- ◆ To remove a mirror from a tier of a file system, enter the following:

```
Storage> tier rmmirror fs_name
```

where *fs_name* specifies the name of the tiered file system from which you want to remove a mirror.

For example:

```
Storage> tier rmmirror fs1
```

This command provides another level of detail for the remove mirror operation. You can use the command to specify which mirror you want to remove by specifying the pool name or disk name.

The disk must be part of a specified pool.

To remove a mirror from a tier spanning a specified pool or disk

- ◆ To remove a mirror from a tier that spans a specified pool or disk, enter the following:

```
Storage> tier rmmirror fs_name [pool_or_disk_name]
```

fs_name Specifies the name of the file system from which to remove a mirror. If the specified file system does not exist, an error message is displayed.

pool_or disk_name Specifies the pool or disk from which the mirror of the tiered file system spans.

The syntax for the `Storage> tier rmmirror` command is the same for both pool and disk. If you try to remove a mirror using `Storage> fs rmmirror fs1 abc`, Veritas Access first checks for the pool with the name `abc`, then Veritas Access removes the mirror spanning on that pool. If there is no pool with the name `abc`, then Veritas Access removes the mirror that is on the `abc` disk. If there is no disk with the name `abc`, then an error message is displayed.

Listing all of the files on the specified tier

You can list all of the files that reside on either the primary tier or the secondary tier.

Note: If the tier contains a large number of files, it may take some time before the output of this command is displayed.

To list all of the files on the specified tier

- ◆ To list all of the files on the specified tier, enter the following:

```
Storage> tier listfiles fs_name {primary|secondary}
```

where **fs_name** indicates the name of the tiered file system from which you want to list the files. You can specify to list files from either the primary or the secondary tier.

Displaying a list of SmartTier file systems

You can display a list of SmartTier file systems using the `Storage> fs list` command.

About tiering policies

Each tier can be assigned a policy.

The tiering policies include:

- Specify on which tier (primary or secondary) the new files get created.
- Relocate files from the primary tier to the secondary tier based on any number of days of inactivity of a file.
- Relocate files from the secondary tier to the primary tier based on the access temperature of the file.
- Prune files on the secondary tier based on the number of days of inactivity of a file.

Prune policies are related to the secondary tier. Prune policies are not related to the cloud tier.

About configuring the policy of each tiered file system

You can configure the policy of each tiered file system.

Table 24-1 Tier policy commands

Command	Definition
tier policy list	Displays the policy for each tiered file system. You can have one policy for each tiered file system. See “Configuring the policy of each tiered file system” on page 354.
tier policy modify	Modifies the policy of a tiered file system. The new files are created on the primary tier. If a file has not been accessed for more than seven days, the files are moved from the primary tier to the secondary tier. If the access temperature is more than five for of the files in the secondary tier, these files are moved from the secondary tier to the primary tier. The access temperature is calculated over a three-day period. See “Configuring the policy of each tiered file system” on page 354.

Table 24-1 Tier policy commands (*continued*)

Command	Definition
tier policy prune	<p>Specifies the prune policy of a tiered file system.</p> <p>Once files have aged on the secondary tier, the prune policy can be set up to delete those aged files automatically.</p> <p>The sub-commands under this command are:</p> <ul style="list-style-type: none"> ■ tier policy prune list ■ tier policy prune modify ■ tier policy prune remove <p>See “Configuring the policy of each tiered file system” on page 354.</p>
tier policy run	<p>Runs the policy of a tiered file system.</p> <p>See “Configuring the policy of each tiered file system” on page 354.</p>
tier policy remove	<p>Removes the policy of a tiered file system.</p> <p>See “Configuring the policy of each tiered file system” on page 354.</p>

Best practices for setting relocation policies

Consider the following relocation policy. The following clauses for relocating the files are:

- Clause 1: If the files on the primary tier are not accessed for 10 days, relocate the files to the secondary tier.
- Clause 2: If the Access Temperature of the files on the secondary tier is more than 100 in the last 15 days, then relocate the files to the primary tier.

```
Storage> tier policy list
FS                               Create on    Days    MinAccess Temp    PERIOD
=====
non_pgr                         primary      10      100                15
```

Setting such policies where the "PERIOD" is more than the "Days" may result in files moving between the tiers after running the `Storage> tier policy run` command. For example, if a file `a.txt` was being used a lot between 1-5 days, and the number of inputs/outputs rises to 3000. After the fifth day, the file was not used for 10 days and the `Storage> tier policy run` command was issued. The file `a.txt` has the Access Temp as $3000/15$, which is equal to 200. As the file has not been used in the last ten days, the file is moved to the secondary tier. If the `Storage>`

`tier policy run` command is run again, the file moves to the primary tier, as the Min Access Temperature is more than 100.

A best practice is to keep the period for calculating the Minimum Access Temperature to lower than the number of days for checking the access age.

Relocating a file or directory of a tiered file system

The `Storage> tier relocate` command relocates a file or directory from a secondary tier to a primary tier. This command does not relocate the NDS (Named Data Stream) files that also include extended attributes to the primary tier.

Note: Relocation is not possible if the primary tier of the file system is full. No error message displays.

To relocate a file or directory

- ◆ To relocate a file or directory, enter the following:

```
Storage> tier relocate fs_name dirPath
```

fs_name	The name of the tiered file system from which you want to relocate a file or directory. The relocation of the file or directory is done from the secondary tier to the primary tier.
dirPath	Enter the relative path of the directory (<code>dirPath</code>) you want to relocate. Or enter the relative path of the file (<code>file_path</code>) that you want to relocate.

Displaying the tier location of a specified file

To display the tier location of a specified file

- ◆ To display the tier location of a specified file, enter the following:

```
Storage> tier mapfile fs_name file_path
```

fs_name	Specifies the name of the file system for which the specified file on the tiered file system resides. If the specified file system does not exist, an error message is displayed.
file_path	Specifies the tier location of the specified file. The path of the file is relative to the file system.

For example, to show the location of `a.txt`, which is in the root directory of the `fs1` file system, enter the following:

```
Storage> tier mapfile fs1 /a.txt
```

About configuring schedules for all tiered file systems

The `tier schedule` commands display, modify, and remove the tiered file systems.

Table 24-2 Tier schedule commands

Command	Definition
tier schedule modify	Modifies the schedule of a tiered file system. See “Configuring schedules for tiered file systems” on page 366.
tier schedule list	Displays the schedules for all tiered file systems. You can have one schedule for each tiered file system. You cannot create a schedule for a non-existent or a non-tiered file system. See “Configuring schedules for tiered file systems” on page 366.
tier schedule remove	Removes the schedule of a tiered file system. See “Configuring schedules for tiered file systems” on page 366.

Configuring schedules for tiered file systems

To modify the schedule of a tiered file system

- ◆ To modify the schedule of a tiered file system, enter the following:

```
Storage> tier schedule modify fs_name minute hour
day_of_the_month month day_of_the_week
node_name
```

The default node name is the master node.

Note: If a previous schedule operation is still running, a new schedule is not created until the previous schedule operation is completed.

fs_name	Specifies the file system where the schedule of the tiered file system resides. If the specified file system does not exist, an error message is displayed.
minute	<p>This parameter may contain either an asterisk, (*), which implies "every minute," or a numeric value between 0-59.</p> <p>You can enter */(0-59), a range such as 23-43, or only the *.</p>
hour	<p>This parameter may contain either an asterisk, (*), which implies "run every hour," or a number value between 0-23.</p> <p>You can enter */(0-23), a range such as 12-21, or only the *.</p>
day_of_the_month	<p>This parameter may contain either an asterisk, (*), which implies "run every day of the month," or a number value between 1-31.</p> <p>You can enter */(1-31), a range such as 3-22, or only the *.</p>
month	<p>This parameter may contain either an asterisk, (*), which implies "run every month," or a number value between 1-12.</p> <p>You can enter */(1-12), a range such as 1-5, or only the *. You can also enter the first three letters of any month (must use lowercase letters).</p>
day_of_the_week	This parameter may contain either an asterisk (*), which implies "run every day of the week," or a numeric value between 0-6. The number 0 is interpreted as Sunday. You can also enter the first three letters of the week (must use lowercase letters).

node_name	Specifies the node on which the schedule of the tiered file system will be run. When creating a schedule for a tiered file system, if you do not input a node name, the schedule will be run on the master node. If you specify a node name, the schedule will be run on the specified node.
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

To display schedules for all tiered file systems

- ◆ To display schedules for all tiered file systems, enter the following:

```
Storage> tier schedule list [fs_name]
```

where *fs_name* indicates the name of the tiered file system for which you want to run a policy.

To remove the schedule of a tiered file system

- ◆ To remove the schedule of a tiered file system, enter the following:

```
Storage> tier schedule remove fs_name
```

where *fs_name* is the name of the tiered file system from which you want to remove a schedule.

Displaying the files that may be moved or pruned by running a policy

Before a policy runs, you can display a list of the files that the policy may move or prune. This feature is useful as a "what if" type of analysis. The command does not physically move any file blocks.

To display a list of files that may be moved or pruned by running a policy

- ◆ To display a list of files that may be moved or pruned by running a policy, enter the following:

```
Storage> tier query fs_name
```

where *fs_name* is the name of the tiered file system for which you want to display the list.

Allowing metadata information on the file system to be written on the secondary tier

The `Storage> tier allowmetadata yes` command allows the metadata information on the specified file system to be written on the secondary tier as well. By default, the secondary tier is not configured for storing metadata information on the file system. Tiers configured with this option show `metaOK` in the column `SECONDARY TIER` of the `Storage> fs list` command output.

To allow metadata information on the file system to be written on the secondary tier

- ◆ To allow metadata information on the file system to be written on the secondary tier, enter the following:

```
Storage> tier allowmetadata yes fs_name
```

where `fs_name` is the name of the file system where metadata information can be written on the secondary tier.

Restricting metadata information to the primary tier only

The `Storage> tier allowmetadata no` command restricts the metadata information to the primary tier only. If the primary tier gets full, the write operations to the secondary tier are not served as the metadata updates. They are restricted to the primary tier only.

To restrict metadata information to the primary tier only

- ◆ To restrict metadata information to the primary tier only, enter the following:

```
Storage> tier allowmetadata no fs_name
```

where `fs_name` is the name of the file system where the metadata information is restricted to the primary tier only.

Removing a tier from a file system

The `Storage> tier remove` command removes a tier from the file system and releases the storage that is used by the file system back to the storage pool. All the files on the secondary tier are relocated to the primary tier. The file system must be online when using the `Storage> tier remove` command.

Note: If the storage tier to be removed contains any data residing on it, then the tier cannot be removed from the file system.

Note: Ensure that you remove the policy first by running the `Storage> tier policy remove` command prior to running the `Storage> tier remove` command.

See [“Configuring the policy of each tiered file system”](#) on page 354.

To remove a tier from a file system

- ◆ To remove a tier from a file system, enter the following:

```
Storage> tier remove fs_name
```

where *fs_name* specifies the name of the tiered file system that you want to remove.

Configuring SmartIO

This chapter includes the following topics:

- [About SmartIO for solid-state drives](#)
- [About configuring SmartIO](#)
- [About SmartIO read caching for applications running on Veritas Access file systems](#)
- [Setting up SmartIO read caching for Veritas Access](#)
- [About SmartIO writeback caching for applications running on Veritas Access file systems](#)
- [Tuning the writeback caching](#)
- [Flushing dirty data from a writeback cache area](#)
- [Setting up SmartIO writeback caching for VxFS file systems](#)
- [Verifying the VxFS cache area and monitoring the caching](#)
- [Setting the caching mode](#)
- [Customizing the caching behavior](#)
- [Viewing the caching statistics for a cache area](#)

About SmartIO for solid-state drives

Solid-state drives (SSDs) are devices that do not have spinning disks. Today's solid-state technologies, such as DRAM and NAND flash, provide faster data access, are more efficient, and have a smaller footprint than traditional spinning disks. The data center uses solid-state technologies in many form factors: in-server, all flash arrays, all flash appliances, and mixed with traditional HDD arrays. Each form factor

offers a different value proposition. SSDs also have many connectivity types: PCIe, FC, SATA, and SAS.

Due to the current cost per gigabyte of SSD devices, the best value of SSDs is not as high capacity storage devices. The benefit of adopting SSDs is to improve performance and reduce the cost per I/O per second (IOPS). Data efficiency and placement is critical to maximizing the returns on any data center's investment in solid state.

The SmartIO feature of Veritas Access enables data efficiency on your SSDs through I/O caching. Using SmartIO to improve efficiency, you can optimize the cost per IOPS. SmartIO does not require in-depth knowledge of the hardware technologies underneath. SmartIO uses advanced, customizable heuristics to determine what data to cache and how that data gets removed from the cache. The heuristics take advantage of Veritas Access' knowledge of the characteristics of the workload.

SmartIO uses a cache area on the target device or devices. The cache area is the storage space that SmartIO uses to store the cached data and the metadata about the cached data. To start using SmartIO, you can create a cache area with a single command, while the application is online.

When the application issues an I/O request, SmartIO checks to see if the I/O can be serviced from the cache. As applications access data from the underlying volumes or file systems, certain data is moved to the cache based on the internal heuristics. Subsequent I/Os are processed from the cache.

SmartIO supports read and write caching for the VxFS file systems that are mounted on VxVM volumes, in several caching modes and configurations.

See [“About SmartIO read caching for applications running on Veritas Access file systems”](#) on page 371.

See [“About SmartIO writeback caching for applications running on Veritas Access file systems”](#) on page 373.

About configuring SmartIO

The `SmartIO` commands control the SmartIO caching functionality of the Veritas Access software.

About SmartIO read caching for applications running on Veritas Access file systems

Veritas Access supports read caching on solid-state drives (SSDs) for applications running on Veritas Access file systems. In this scenario, application reads are

satisfied from the cache whenever possible. As the application accesses the file system, the file system loads data from the disk into the cache. Application writes go to the disk in the usual way. With each write, the file system synchronizes the cache to ensure that applications never see stale data. If a cache device fails, a file that is cached in read mode is completely present on the disk. Therefore, the cache failure does not affect the application I/Os for the file and the application I/Os continue without interruption.

By default, the cache area is enabled for caching. All file systems on the system are cached unless you explicitly disable caching for that file system. You do not need to explicitly enable caching on a file system.

Setting up SmartIO read caching for Veritas Access

In read mode, the SmartIO feature caches the file system read I/Os. To set up SmartIO for read caching for a file system, simply create the cache area.

Setting up SmartIO read caching

- 1 For each node, to view a list of devices available to use for the SmartIO cache area, use the following command:

```
SMARTIO> device list node_name
```

Where:

node_name specifies the cluster node.

- 2 Create the cache area on the SSD device, using the following command.

```
SMARTIO> cache create nodename ssddisk size  
fstype cachename
```

Where:

nodename specifies the cluster node on which to create the SmartIO cache.

ssddisk specifies a device to use for the cache area.

size specifies the SmartIO cache size.

fstype specifies the file system type (default/reserve).

cachename specifies the cache name.

3 The cache area is set to read mode by default.

When a cache area is deleted or brought offline, the caching mode is not removed from the file system mount options. If you create or bring online a new cache area, the cache area inherits the existing caching mode.

If the file system mount option was previously set to writeback, you must explicitly change the caching mode as follows:

```
SmartIO> fs cachemode read fs_name
```

4 If required, you can further customize the caching behavior.

About SmartIO writeback caching for applications running on Veritas Access file systems

Veritas Access supports writeback caching on solid-state drives (SSDs) for applications running on Veritas Access file systems. In this scenario, application reads and writes are satisfied from the cache whenever possible.

SmartIO provides write caching in the writeback mode. In writeback mode, an application write returns success after the data is written to the SmartIO cache, which is usually on an SSD. At a later time, SmartIO flushes the cache, which writes the dirty data to the disk. Writeback caching expects to improve the latencies of synchronous user data writes. Write order fidelity is not guaranteed while flushing the dirty data to the disk.

Writeback caching is superset of read caching. When writeback caching is enabled, read caching is implicitly enabled. Reads are satisfied from the cache if possible, and the file system transparently loads file data into the cache. Both read and writeback caching may be enabled for the same file at the same time.

The writeback caching mode gives good performance for writes, but also means that the disk copy may not always be up to date. If a cache device fails, a file that is cached in writeback mode may not be completely present on the disk. SmartIO has a mechanism to flush the data from the cache device when the device comes back online. Veritas Access provides additional protection from data loss with cache reflection. Cache reflection is enabled by default.

Writeback caching requires a cluster with exactly two nodes. Writeback caching cannot be enabled if the cluster has more than two nodes or if the cluster has a single node.

When writeback caching is enabled, SmartIO mirrors the writeback data at the file system level to the other node's SSD cache. This behavior, called cache reflection, prevents loss of writeback data if a node fails. If a node fails, the other node flushes

the mirrored dirty data of the lost node as part of reconfiguration. Cache reflection ensures that writeback data is not lost even if a node fails with pending dirty data.

After writeback caching is enabled on the mount point, the qualified synchronous writes in that file system are cached. SmartIO determines if a write qualifies for writeback caching, using criteria such as the following:

- The write request must be PAGESIZE aligned (multiple of 4 KB).
- The write request is not greater than 2 MB.
- The file on which the writes are happening is not mapped.
- Writeback caching is not explicitly disabled by the administrator.
- Writeback caching is not qualified if the cluster has more than two nodes.

You can also customize which data is cached, by adding advisory information to assist the SmartIO feature in making those determinations.

Tuning the writeback caching

When writeback caching is enabled, any data that is read from the disk is cached, unless the file is explicitly marked for "no caching" or if the cache is full. For writes, certain writes cause the data to be cached. You can load a file to speed up the application. Pinning a file in the cache ensures that the data does not get evicted. If some data is already cached, and that portion of the disk is overwritten, then SmartIO also writes the new data to the cache device to ensure that the cached data remains up to date.

You can use the following tunable parameters to adjust the size of the cache and how long data is held in the cache.

Setting the maximum space used for dirty data per node

When writeback is enabled, you can configure a maximum for the space used for dirty data per node. By default, there is no maximum. If you configure a maximum, you must allow at least 512 MB for each file system or cluster file system. For a cluster file system, the space required for data reflection is not included in the maximum.

To set the writeback size

- 1 Run the following command to configure the maximum. For a cluster file system, run the command on each node of the cluster to make the setting cluster wide.

```
SmartIO> cache wb_size nodename cachename cache_size
```

Where:

nodename specifies the cluster node.

cachename specifies the name of the cache.

cache_size specifies the size of the cache.

- 2 For the changed value to take affect, run the following command.

```
SMARTIO> fs enable target1
```

- 3 The writeback size is shown as Writeback Cache Use Limit, when you display the cache statistics. To display the cache statistics:

```
SMARTIO> cache stat
```

See [“Viewing the caching statistics for a cache area”](#) on page 379.

Flushing dirty data from a writeback cache area

With SmartIO, dirty data in the cache is automatically flushed to the disk during normal operations. The dirty data is flushed when the file system is unmounted, or during other operations that require a file system freeze. The dirty data is also flushed periodically at intervals. You can control the interval by configuring the tunable parameters.

See [“Tuning the writeback caching”](#) on page 374.

Disabling writeback caching for a file also flushes any writeback dirty data for that file.

In some cases, you may want to manually trigger flushing of the dirty data from the cache to the disk. For example, to ensure data consistency, you would flush the cache before you create an array level snapshot.

You can manually trigger flushing of the dirty data using the following command.

```
SmartIO> fs flush target1
```

Setting up SmartIO writeback caching for VxFS file systems

In writeback mode, the SmartIO feature caches the VxFS file system read and write I/Os. To set up SmartIO for writeback caching for a VxFS file system, create the cache area and set the caching mode to writeback mode.

Writeback caching requires a cluster with exactly two nodes.

To set up SmartIO writeback caching for VxFS file systems

- 1 For each node, to view a list of devices available to use for the SmartIO cache area, use the following command:

```
SMARTIO> device list node_name
```

Where:

node_name specifies the cluster node.

- 2 Create the VxFS cache area on the SSD device, using the following command.

```
SMARTIO> cache create nodename ssddisk size  
           fstype cachename
```

Where:

nodename specifies the cluster node on which to create the SmartIO cache.

ssddisk specifies a device to use for the cache area.

size specifies the SmartIO cache size.

fstype specifies the file system type (default/reserve).

cachename specifies the cache name.

- 3 Set the caching mode for the file system to writeback, using the following command.

```
SmartIO> fs cachemode writeback fs_name
```

- 4 If required, you can further customize the caching behavior.

See [“Customizing the caching behavior”](#) on page 378.

See [“Tuning the writeback caching”](#) on page 374.

Verifying the VxFS cache area and monitoring the caching

After the SmartIO feature is configured, you can verify that the cache area is present and that caching is occurring.

To verify and monitor the cache area

- 1 Use the following command to display information about the cache areas on the system.

```
SmartIO> cache list
```

- 2 To display information about a specific file:

```
SmartIO> file list target1
```

- 3 To see statistics on the cache usage, use the following command:

```
SmartIO> fs stat target1
```

The output displays statistics for the cached data.

See [“Viewing the caching statistics for a cache area”](#) on page 379.

Setting the caching mode

You can set the caching mode of a file system. For a VxFS cache area, the caching mode determines what kind of caching is performed for the specified mount point. The mode can be `nocache`, `read`, or `writeback`. The default mode is `read`.

When a cache area is deleted or brought offline, the caching mode of the cache area is not changed. If you create or bring online a new cache area, the cache area inherits the existing caching mode. If a different caching mode is required, you must explicitly change the caching mode.

To change the caching mode of a file system

- ◆ To change the caching mode of a file system, use the following command.

```
SmartIO> fs cachemode {nocache|read|writeback} file system
```

Customizing the caching behavior

By default, SmartIO caches the file data based on the workload. SmartIO loads portions of files into the cache based on I/O access. When the cache area fills, data may be evicted to make room for caching new data. SmartIO uses criteria such as frequency of access to evict data. While the data is in the cache, the subsequent I/Os to that file data are satisfied from the cache. If the data is evicted, any subsequent I/O request is served from the primary storage. SmartIO may then cache the data again.

To maximize the use of the cache, you can customize the caching behavior to control when files are loaded or evicted from the cache. You can customize the caching behavior, using the following operations:

- The `load` operation preloads files into the cache before the I/O accesses the files. The files are already in the cache so that the I/Os return more quickly. The files are loaded asynchronously in the background.
- The `pin` operation prevents the files from being evicted from the cache. You can pin commonly used files so that SmartIO does not evict the files and later need to cache the files again. A pinned file is kept in the cache indefinitely, until it is deleted or explicitly unpinned.
- The `unpin` operation removes files from the pinned state. The `unpin` operation does not cause the file to be immediately evicted. SmartIO considers the file for eviction in the same way as any other file, when space is required in the cache.

For each of these operations, you can specify files individually, or specify a directory name to affect all of the files in a directory.

To load a file or directory

- ◆ To load a file or directory to the cache, specify the file name or the directory name to the following command.

```
SmartIO> file load {file|dir}
```

Where:

file is the full path name of the file to be loaded.

dir is the directory name. If you specify a directory name, all of the files in the directory are loaded.

To pin a file or directory

- ◆ To pin a file or directory to the cache, specify the file name or directory name to the following command.

```
SmartIO> file pin {file|dir}
```

Where:

file is the full path name of the file to be pinned.

dir is the directory name. If you specify a directory name, all of the files in the directory are pinned.

To unpin a file or directory

- ◆ To unpin a file or directory to the cache, specify the file name to the following command.

```
SmartIO> file unpin {file|dir}
```

Where:

file is the full path name of the file to be unpinned.

dir is the directory name. If you specify a directory name, all of the files in the directory are unpinned.

Viewing the caching statistics for a cache area

For a VxFS cache area, the statistics do not change after you unmount and mount the file systems. For a cluster file system, the statistics do not change after you reboot the cluster nodes.

To view the caching statistics for a VxFS cache area

- ◆ Use the following command:

```
SMARTIO> cache stat
```

Configuring episodic replication

This chapter includes the following topics:

- [About Veritas Access episodic replication](#)
- [How Veritas Access episodic replication works](#)
- [Starting Veritas Access episodic replication](#)
- [Setting up communication between the source and the destination clusters](#)
- [Setting up the file systems to replicate](#)
- [Setting up files to exclude from an episodic replication unit](#)
- [Scheduling the episodic replication](#)
- [Defining what to replicate](#)
- [About the maximum number of parallel episodic replication jobs](#)
- [Managing an episodic replication job](#)
- [Replicating compressed data](#)
- [Displaying episodic replication job information and status](#)
- [Synchronizing an episodic replication job](#)
- [Behavior of the file systems on the episodic replication destination target](#)
- [Accessing file systems configured as episodic replication destinations](#)
- [Creating a recovery point objective \(RPO\) report](#)

- [Episodic replication job failover and fallback](#)

About Veritas Access episodic replication

The Veritas Access episodic replication solution provides high performance, scalable data replication and is ideal for use as a content distribution solution, and for use to create hot standby copies of important data sets.

Veritas Access episodic replication lets you asynchronously replicate a file system from one node in a source cluster to another node in a destination cluster at regularly timed intervals. This allows for content sharing, replication, and distribution.

The Veritas Access episodic replication functionality allows episodic replication with a minimum timed interval update of 15 minutes and no set maximum. Unlike many replication solutions, Veritas Access episodic replication also allows the destination file system to be online for reads while replication is active.

Major features of Veritas Access episodic replication include:

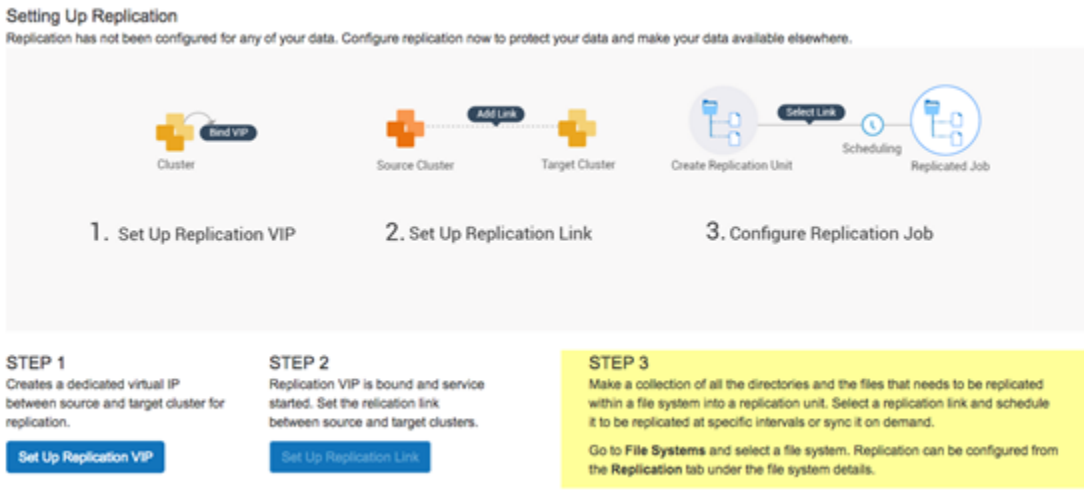
- Online access (read-only) to replicated data.
- Immediate read/write access to destination replicated data in the unlikely event that the source file system goes offline for a sustained period of time.
- Load balancing across replication links.
- Transport failover of episodic replication service from one node to another.
- Unlimited simultaneous episodic replication operations.

The Veritas Access episodic replication feature is designed to copy file systems only between Veritas Access clusters.

Note: The Veritas Access episodic replication feature does not support user modifications to the target file system if episodic replication is configured. It is also not supported for a scale-out file system.

[Figure 26-1](#) describes the workflow for configuring episodic replication between two Veritas Access clusters.

Figure 26-1 Episodic replication workflow



How Veritas Access episodic replication works

Veritas Access episodic replication is an incremental episodic replication service that runs on top of the Cluster File System that is used by Veritas Access which is, in turn, based on the Veritas File System (VxFS). Veritas Access episodic replication uses two file system specific features: File Change Log (FCL) and Storage Checkpoint services, to retrieve file changes between replication periods.

For a given period, the FCL records every change made to the file system. By scanning the FCL, Veritas Access episodic replication quickly identifies the file(s) that have changed and generates the modified file list. This avoids the expensive file system scanning that is normally associated with file-based replication, and which typically results in sub-optimal performance.

Next, Veritas Access episodic replication uses VxFS Storage Checkpoint's metadata comparison feature to retrieve the modified extent list of each changed file. It does not need to access the file data.

The Veritas Access episodic replication transport layer works in conjunction with, and interfaces to the well-known rsync remote file synchronization tool. Using this existing network transportation program makes the network configuration much easier in the enterprise domain: the Secure Socket Shell (SSH) port (22) required by rsync is opened by default on almost all enterprise firewalls. rsync is also a reliable solution for a low bandwidth or unreliable link environment.

Note: Veritas Access uses the rsync protocol to provide transportation of Veritas Access episodic replication encapsulated files. The use of rsync is not exposed in Veritas Access, and cannot be administered outside of the Veritas Access episodic replication feature set.

Starting Veritas Access episodic replication

This section lists the specific commands that are needed to run Veritas Access episodic replication on your clusters.

Ensure the following before starting episodic replication:

- Before you set up your clusters for episodic replication, you must first identify which is the source cluster and which is the destination cluster. All of the commands are performed on the source cluster first.
- Make sure both the source cluster and the destination cluster have the same version of Veritas Access.
- To use Veritas Access episodic replication, you must first create an online file system on the Veritas Access source cluster and an online file system on the Veritas Access destination cluster.
- Assign a virtual IP (VIP) address to both the source and the destination clusters. The Veritas Access episodic replication service requires VIP addresses not already in use for the two clusters to communicate.

To start Veritas Access episodic replication on the source cluster

- 1 To bind a virtual IP address for the episodic replication service on the source cluster, enter the following:

```
Replication> episodic config bind ip_addr [device] [netmask]
```

ip_addr Virtual IP address for the episodic replication service on the destination cluster.

device The public network interface name that you want the episodic replication IP address to use.

netmask Netmask for the episodic replication IP address.

- 2 To start the episodic replication service, enter the following on the source node:

```
Replication> episodic service start [nodename]
```

nodename The name of the node in the local cluster where you want to start the episodic replication service.

- 3 To check the status of the episodic replication service, enter the following:

```
Replication> episodic service status
```

- 4 To confirm the IP address is up and running, enter the following:

```
Replication> episodic config show ip
```

The definitions of the headings are as follows:

Note: Alternately, you can use the `Network> ip addr show` command to confirm that the IP address is up and running.

To start Veritas Access episodic replication on the destination cluster

- 1 To bind a virtual IP address for the episodic replication service on the destination cluster, enter the following:

```
Replication> episodic config bind ip_addr [device] [netmask]
```

ip_addr Virtual IP address for the episodic replication service on the source cluster.

device The public network interface name that you want the episodic replication IP address to use.

netmask Netmask for the episodic replication IP address.

- 2 To start the episodic replication service, enter the following on the destination node:

```
Replication> episodic service start [nodename]
```

nodename The name of the node in the local cluster where you want to start the episodic replication service.

- 3 To check the status of the episodic replication service, enter the following:

```
Replication> episodic service status
```

- 4 To confirm that the IP address is up and running, enter the following:

```
Replication> episodic config show ip
```

You next need to set up communication between the source and the destination clusters.

Setting up communication between the source and the destination clusters

You need to set up communication between your source and your destination clusters.

Make sure that you already created an online file system on the Veritas Access source cluster and an online file system on the Veritas Access destination cluster.

Veritas Access episodic replication authentication strategy is based on RSA-key authentication, and both the source and the destination clusters have to export their

episodic replication public keys. The source cluster imports the destination cluster's public key and the destination cluster imports the source cluster's public key.

After you have determined which two Veritas Access clusters to use, you need to authenticate them.

The `Replication> episodic config` commands must be executed in a specific order.

- Use the `Replication> episodic config del_keys` **after** the `Replication> episodic config deauth` command, or it fails.
- You can only run the `Replication> episodic config unbind` command (to unbind the virtual IP) **after** you have run the `Replication> episodic service stop` command.
- You need to run the `Replication> episodic config bind` command (to bind the virtual IP) **before** you can run the `Replication> episodic service start` command.
- You need to run the `Replication> episodic config export_keys` and `Replication> episodic config import_keys` to export and import the keys of both the source and the destination clusters.
- You can only run the `Replication> episodic config auth` command **after** both the source and destination have imported each others keys.
- You need to run the `Replication> episodic config auth` command to create a link from every cluster to any remaining cluster that is used for episodic replication irrespective of their role as a source or a destination cluster.

After the source and the destination clusters have successfully imported each other's public keys, you need to run the `Replication> episodic config auth` command on the source cluster to complete the authentication between the two clusters. This command checks the two-way communication between the source and the destination cluster, and authenticates the clusters allowing the Veritas Access episodic replication service to begin.

Note: The `Replication> episodic config auth` command must be executed from the source cluster.

This section provides a walk-through for the creation and export/import of these encrypted keys for both the source and the destination cluster.

Note: Without the correct authentication of the source and the destination encryption keys, Veritas Access episodic replication does not function correctly.

To export the source cluster's key to the destination cluster

- 1 To export the source cluster's key to the destination cluster, enter the following:

```
Replication> episodic config export_keys [URL]
```

URL The location you want to copy the public keys to.

If you do not want to enter a URL, you can copy the output from the `Replication> episodic config export_keys` command into the `Replication> episodic config import_keys` command at the destination cluster.

By default, the output is displayed to your computer screen.

The SCP and FTP protocols are supported.

- 2 To import the source cluster's key to the destination cluster, enter the following:

```
Replication> episodic config import_keys [URL/keyfile]
```

URL The location you want to copy the public keys from.

keyfile The file name of the key that is generated by the export.

If you did not enter a URL during the `Replication> episodic config export_keys` command, you can cut and paste the output and enter it into the `Replication> episodic config import_keys` command.

- 3 To verify that the key has been imported correctly, enter the following:

```
Replication> episodic config show
```

To export the destination cluster's key to the source cluster

- 1** To export the destination cluster's key to the source cluster, enter the following:

```
Replication> episodic config export_keys [URL]
```

URL The location you want to copy the public keys to.

The SCP and FTP protocols are supported.

If you do not want to enter a URL, you can cut and paste the output from the `Replication> episodic config export_keys` command to the `Replication> episodic config import_keys` command. By default, the output is displayed to your computer screen.

- 2** To import the destination cluster's key to the source cluster, enter the following:

```
Replication> episodic config import_keys [URL/keyfile]
```

URL Enter the URL of the location you want to copy the public keys from.

keyfile Enter the file name of the key that is generated by the export.

If you did not enter a URL during the `Replication> episodic config export_keys` command, you can cut and paste the output and enter it into the `Replication> episodic config import_keys` command.

- 3** To verify that the key has been imported correctly, enter the following:

```
Replication> episodic config show
```

To authenticate source cluster and destination clusters for episodic replication

- 1 This command should be executed on the source cluster as well as on the destination cluster. To authenticate the public keys on the source cluster and the destination clusters, enter the following:

```
Replication> episodic config auth conIP link_name
```

conIP Enter the destination cluster console IP address.

link_name Both the source cluster and the destination cluster need to be assigned a unique identifier (name). This identifier is used to identify the link that is established between the source and the destination clusters. You can use the link name instead of the virtual IP addresses of the source and the destination clusters when using the other episodic replication commands. For example:
Pune_Shanghai.

- 2 To confirm the authentication, enter the following:

```
Replication> episodic config show
```

Note: These steps must be executed on the destination side cluster to authenticate the public keys on the source and the destination cluster.

Once you have configured the clusters and links, you need to set up the file systems you want to replicate.

Setting up the file systems to replicate

You need to set up the file systems you want to replicate using the `Replication> episodic repunit` commands. The `Replication> episodic repunit` commands let you define the type of data that you replicate from the source cluster to the destination cluster. All files and folders belonging to a episodic replication unit are replicated together from the source cluster to the destination cluster.

Note: The maximum number of episodic replication units supported in Veritas Access episodic replication is 128.

Make sure that you already set up communication between your source and the destination clusters.

See [“Setting up communication between the source and the destination clusters”](#) on page 385.

An episodic replication unit is defined as an ordered set of entries, where each entry is one of the following:

- A single file system
- A single subdirectory
- A single file

Note: The episodic replication source has to be one of the entry types shown. It cannot be a snapshot or a Storage Checkpoint (ckpt).

Veritas Access episodic replication requires that the source and the destination episodic replication units of a job definition have the same type of ordered entries, that is, every entry pair (one entry from the source and one entry from the destination episodic replication unit) must be of a similar type.

Both can be files, or both can be directories, as shown in the following example:

Replication unit Name	Replication unit Entries
=====	=====
ru1	fs1,fs2/dir1,fs2/f1
ru2	fs4,fs6/dir2,fs5/f2

The entry is identified by the file system name, optionally followed by a slash '/', followed by the path of the directory or the file inside the file system. Member entries are ordered inside a episodic replication unit and such ordering information is used to determine the episodic replication entity pair mapping from the source episodic replication unit to the destination episodic replication unit.

Note: Make sure that the paths in the destination episodic replication unit exist in the destination cluster.

Note: The commands in this section apply only to the source episodic replication unit.

To create an episodic replication unit

- 1 From the source cluster, to create an episodic replication unit, enter the following:

```
Replication> episodic repunit create repunit_name  

               repunit_entry[,repunit_entry,...]
```

repunit_name The name of the episodic replication unit you want to create.

repunit_entry The file system file, file, folder, or directory.

Note: Destination episodic replication units should be created only at the source cluster using the `Replication> episodic repunit create` command.

- 2 To confirm the creation of the episodic replication unit, enter the following:

```
Replication> episodic repunit show verbose
```

You can use the `Replication> episodic repunit add_entry`, `Replication> episodic repunit modify_entry`, `Replication> episodic repunit remove_entry`, and `Replication> episodic repunit destroy` commands to manage your episodic replication units.

Note: The `Replication> episodic repunit modify_entry`, `Replication> episodic repunit remove_entry`, and `Replication> episodic repunit destroy` operations are not allowed for the episodic replication units that are included in any job definitions.

Setting up files to exclude from an episodic replication unit

Once you have set up the files systems you want to replicate, you can define a set of directories or files to exclude from an episodic replication unit. This step is optional. The `exclunit` entry has higher priority over the `repunit` entry. If any file name matches the `exclunit` entry, the file is not replicated to the target.

To work with exclusion units:

- Use the `Replication> episodic exclunit create` command to name the excluding unit and configure the directories and files you want to exclude from an episodic replication. The excluding unit you create can be used in multiple

episodic replication jobs. A single excluding unit can span across multiple directories.

- Use the `Replication> episodic job exclude` command to add the excluding unit to a episodic replication job. You cannot add an excluding unit to a job that is active. You must disable the job first.
- You can use the following commands: `Replication> episodic exclunit add_entry`, `Replication> episodic exclunit modify_entry`, and `Replication> episodic exclunit remove_entry` to make changes to an excluding unit, provided the excluding unit you want to modify is not included in any job definitions.
- Use the `Replication> episodic job show` command to show which excluding units are configured for a job. Use the `Replication> episodic exclunit show` command to show the names and contents of all excluding units that are defined for the cluster.
- Use the `Replication> episodic exclunit destroy` command to permanently delete the excluding unit. You can only destroy an excluding unit if the excluding unit you want to destroy is not included in any job definitions.

If an episodic replication is defined for a directory, an excluding unit should be a subset of that directory. The excluding unit cannot be the same directory as the episodic replication and it cannot be a parent directory of the episodic replication. For example, if a episodic replication is configured for `fs1/dir1/dir2`, a valid exclusion could be `dir1/dir2/file` or `dir1/dir2/dir3`, but not `/dir1` (the parent directory for the episodic replication).

By default, Veritas Access excludes some common directories and files from all episodic replication units. These directories and files include:

- `lost+found`
- `.placement_policy.xml`
- `quotas`
- `quotas.grp`
- `quotas.64`
- `quotas.grp.64`

In addition, you can use the `Replication> episodic exclunit` commands to specify additional directories and files to exclude.

The directories and files you specify for an excluding unit are applied based on the overall definition of the episodic replication. For example, an episodic replication

job that contains an `fs1` episodic replication unit and an `dir3` excluding unit, replicates all the files in `fs1`, except for the files in `fs1/dir3`.

To create an excluding unit:

- 1 To create an excluding unit, enter the following:

```
Replication> episodic exclunit create exclunit_name  
                                exclunit_entry[,exclunit_entry,...]
```

exclunit_name Enter the name of the excluding unit.

exclunit_entry Enter the comma-separated list of directories and files you want to exclude from an episodic replication.

- 2 To confirm the creation of the excluding unit enter the following:

```
Replication> episodic exclunit show verbose
```

You can use the `Replication> episodic exclunit add_entry`, `Replication> episodic exclunit modify_entry`, `Replication> episodic exclunit remove_entry`, and `Replication> episodic exclunit destroy` commands to manage your excluding units.

Note: The `Replication> episodic exclunit add_entry`, `Replication> episodic exclunit modify_entry`, `Replication> episodic exclunit remove_entry`, and `Replication> episodic exclunit destroy` operations are not allowed for excluding units that are included in any job definitions.

Scheduling the episodic replication

You use the `Replication> episodic schedule` commands to create a schedule for replicating files from the source to the destination cluster.

Veritas Access episodic replication supports periodic replications, where the data gets replicated from the source to the destination cluster at regular intervals as defined by the schedule. Veritas Access episodic replication uses the following parameters to schedule the episodic replication jobs: minute, hour, day-of-the-month, month, and day-of-the-week.

Make sure that you already set up the file systems you want to replicate.

See [“Setting up the file systems to replicate”](#) on page 389.

To create a episodic replication schedule

- ◆ To create an episodic replication schedule, enter the following:

```
Replication> episodic schedule create schedule_name minute
[hour] [day_of_the_month] [month] [day_of_the_week]
```

<i>schedule_name</i>	Specify the name of the schedule to be created.
<i>minute</i>	Enter a numeric value between 0-59, or an asterisk (*), which represents every minute. This variable is not optional.
<i>hour</i>	Enter a numeric value between 0-23, or an asterisk (*), which represents every hour.
<i>day_of_the_month</i>	Schedule the day of the month you want to run the replication. Enter a numeric value between 1-31, or an asterisk (*), which represents every day of the month.
<i>month</i>	Schedule the month you want to run the replication. Enter a numeric value between 1-12, or an asterisk (*), which represents every month. You can also use the names of the month. Enter the first three letters of the month (not case sensitive).
<i>day_of_the_week</i>	Schedule the day of the week you want to run the replication. Enter a numeric value between 0-6, or an asterisk (*), which represents every day of the week. Sunday is interpreted as 0. You can also enter the first three letters of the week (you must use lower case letters).

You can enter an interval (two numbers separated by a hyphen) for the *minute*, *hour*, *day-of-month*, *month*, and *day-of-week*. If you want to run the schedule between 1:00 a.m. and 4:00 a.m., you can enter a value of 1-4 for the hour variable. The range is inclusive

The parameters also accept a set of numbers separated by a comma. For example, 1,3,5,7 or 1-4,5-10.

To display the list of schedules

- ◆ To display the schedule you have set up for episodic replication, enter the following:

```
Replication> episodic schedule show
```

You can also use the `Replication> episodic schedule modify` and `Replication> episodic schedule delete` to manage your episodic replication schedules.

Note: The `Replication> episodic schedule modify` and `Replication> episodic schedule delete` operations are not allowed for the schedules that are included in any job definition.

You next need to define what is replicated.

See [“Defining what to replicate”](#) on page 395.

Defining what to replicate

You use the `Replication> episodic job` commands to set up a job definition. This defined job determines what to replicate and when, using the settings from the previous commands.

Make sure that you created a schedule for replicating files from the source to the destination cluster.

See [“Scheduling the episodic replication”](#) on page 393.

To set up the episodic replication job

- 1 To create an episodic replication job, enter the following:

```
Replication> episodic job create job_name src_repunit tgt_repunit  
link_name schedule_name [evpsn] [metadata_only]
```

<i>job_name</i>	Specify a name for the episodic replication job you want to create.
<i>src_repunit</i>	Specify the source episodic replication unit. The episodic replication unit determines the exact item (such as a file system) that you want to replicate.
<i>tgt_repunit</i>	Specify target episodic replication units.
<i>link_name</i>	Specify the link name used when you ran the <code>Replication> episodic config auth</code> command between the local cluster and the remote cluster. Both the source cluster and the destination cluster need to be assigned a unique identifier (name). This identifier is used to identify the link that is established between the source and the destination clusters. You can use the link name instead of the virtual IP addresses of the source and the destination clusters when using the other episodic replication commands.
<i>schedule_name</i>	Specify the name of the episodic replication schedule you want to apply to the episodic replication job.
<i>evpsn</i>	Enable or disable Enterprise Vault partition secure notifications.
<i>metadata_only</i>	Enable or disable metadata-only episodic replication. This feature is not supported with consistency groups and tunables.

- 2 To add an excluding unit to the job, enter the following command. This step is optional.

```
Replication> episodic job exclude job_name exclunit_name
```

- 3 By default, the job is disabled. To enable the job, enter the following:

```
Replication> episodic job enable job_name
```

- 4 To check if the job was enabled, enter the following:

```
Replication> episodic job show [job_name]
```

About the maximum number of parallel episodic replication jobs

The maximum number of episodic replication jobs is 64, but there are stricter limits on the number of episodic replication jobs that can be running in parallel at the same time. Episodic replication uses a RAM-based file system for storing the transit messages. Each GB of this RAM-based file system can accommodate up to eight parallel running jobs. The default size of this file system depends upon the amount of physical memory of the node on which episodic replication is running. If the physical memory is less than 5 GB, episodic replication limits its maximum usage for storing messages to 1 GB of memory, which means the user can run up to eight episodic replication jobs in parallel at the same time. If the physical memory is between 5 GB to 10 GB, episodic replication limits its maximum usage for storing messages to 2 GB of memory, which means you can run up to 16 episodic replication jobs in parallel. If the physical memory is greater than 10 GB, episodic replication limits its maximum usage for storing messages to 4 GB of memory, which means you can run up to 32 episodic replication jobs in parallel at the same time.

Managing an episodic replication job

You can manage an episodic replication job using the `Replication> episodic job` commands. The commands are required only on the source system.

The `Replication> episodic job enable`, `Replication> episodic job sync`, `Replication> episodic job disable`, `Replication> episodic job abort`, `Replication> episodic job pause`, and `Replication> episodic job resume` commands change the status of an existing episodic replication job.

You can use the `Replication> episodic job modify`, and `Replication> episodic job destroy` commands to modify or destroy an episodic replication job definition.

The `Replication> episodic job enable` command starts episodic replication immediately and initiates episodic replication after every subsequent set frequency

interval. When an episodic replication job is created it is disabled by default, and you must enable the job to start episodic replication.

To enable an episodic replication job

- ◆ To enable an episodic replication job, type the following command:

```
Replication> episodic job enable job_name
```

job_name Specify the name of the episodic replication job you want to enable.

At each frequency interval, a fresh file system Storage Checkpoint is taken and episodic replication is started against the new Storage Checkpoint. If a previous episodic replication run has not completed, a new Storage Checkpoint is not taken and the current run is skipped.

Note: Running the `Replication> episodic job enable` command on a previously aborted episodic replication job automatically restarts the job.

The `Replication> episodic job sync` command lets you start an episodic replication job, but then stops the episodic replication job after one iteration (full or incremental) is complete. You can use this command to recover from the secondary site in the event that the primary file system is completely destroyed. This command can also be used if you want to run an episodic replication job at a predefined time using a script or a cron job.

See [“Synchronizing an episodic replication job”](#) on page 403.

The `Replication> episodic job disable` command drops the episodic replication job from the schedule and waits for any already running iterations to complete. The `Replication> episodic job disable` command disables a job definition which is in one of these states: ENABLED, PAUSED, or FAILED. This process can take some time if the network is slow or if a large amount of data has changed since the last episodic replication run.

To disable an episodic replication job

- ◆ To disable an episodic replication job, type the following command:

```
Replication> episodic job disable job_name
```

job_name Specify the name of the episodic replication job you want to stop.

The `Replication> episodic job abort` command forcefully cancels an episodic replication job even if it is in progress. Aborting an episodic replication job may

leave Storage Checkpoints mounted on the source system and the target file system may be left in an intermediate state.

To abort an episodic replication job

- ◆ To abort an episodic replication job, type the following command:

```
Replication> episodic job abort job_name
```

job_name Specify the name of the episodic replication job you want to abort.

The `Replication> episodic job pause` command immediately stops the episodic replication job. You must use the `Replication> episodic job resume` command to resume the episodic replication job from where it was paused. When episodic replication is resumed, the episodic replication job replicates the set of selected files before pausing the job, and attempts to replicate as much of the latest data as possible. This action allows the customer to have two recovery point objectives (RPO). When the episodic replication job is paused, the episodic replication frequency option is disabled. Once the episodic replication job is resumed, the frequency option resumes for subsequent iterations. The pause and the resume functions let you manage the episodic replication job based on workload requirements.

To pause and resume an episodic replication job

- 1 To pause an episodic replication job, type the following command:

```
Replication> episodic job pause job_name
```

where *job_name* is the name of the episodic replication job you want to pause.

- 2 To resume an episodic replication job, type the following command:

```
Replication> episodic job resume job_name
```

where *job_name* is the name of the episodic replication job you want to resume.

Note: You cannot start or sync a paused job. You can abort a paused job. However, if synchronization is performed on a paused job that has been aborted, the last RPO for the paused job is not available.

The `Replication> episodic job modify` command lets you modify debugging or setting tunables on an episodic replication job definition.

The addition or removal of a filesystem from the source episodic replication unit or the destination episodic replication unit is not supported. To remove a specific filesystem from the episodic replication unit you must destroy the episodic replication

job and recreate the episodic replication job with the new set of file systems in the episodic replication unit. To add a specific filesystem from an existing episodic replication unit, you can either create a new episodic replication job with a new source episodic replication unit and target episodic replication unit, or destroy the episodic replication job and recreate it with the new set of file systems in the episodic replication unit to use the same job name

The `Replication> episodic job modify debug` command lets you enable or disable debugging on a given job.

To modify debugging on an episodic replication job

- ◆ To modify debugging on an episodic replication job definition, enter the following command:

```
Replication> episodic job modify debug job_name on|off
```

job_name Specify the episodic replication job name you want to modify.

The `Replication> episodic job modify tunables` command allows you to modify the job configuration to use multiple network connections (sockets) for replicating data from source to target. In configurations where WAN latency is high, it is recommended to use multiple connections for significantly increased throughput. After the tunables are set for a job, only one job is supported.

To modify tunables on an episodic replication job

- ◆ To modify tunables on an episodic replication job definition, enter the following command:

```
Replication> episodic job modify tunables job_name netconn rw_count
```

job_name Specify the episodic replication job name you want to modify.

netconn Specify the number of connections.

rw_count Specify the number of threads.

The increased number of connections is effective in case of a relatively small number of large files. For large number of small files, full sync performance may be slower with increased number of connections.

The `Replication> episodic job destroy` command destroys a job definition. This command completely removes the specified job from the configuration, cleans up any saved job-related statistics, and removes any Storage Checkpoints. The episodic replication job must be disabled before the job definition can be destroyed.

To destroy an episodic replication job definition

- ◆ To destroy a job definition, enter the following command:

```
Replication> episodic job destroy job_name
```

Where *job_name* is the name of the job definition you want to delete. Make sure that the job is not enabled.

Using the `Replication> episodic job destroy` command with the `force` option removes the local job irrespective of the job state, and all episodic replication units are disassociated from the job. Cluster configurations, which are part of the job, are not modified.

Note: When setting up episodic replication, Veritas does not advise you to make any modifications or deletions on the target side of the file system. In the event that some or all of the target data is modified or deleted, you must re-create the episodic replication job from the source cluster to resume episodic replication services

To re-create an episodic replication job

- 1 To re-create an episodic replication job, you must first delete the job definition. Enter the following command on the source cluster:

```
Replication> episodic job destroy job_name
```

Where *job_name* is the name of the job definition you want to delete. Make sure that the job is not enabled.

- 2 Re-create the job definition:

```
Replication> episodic job create job_name src_repunit tgt_repunit  
link_name schedule_name [evpsn] [metadata_only]
```

You can reuse the source episodic replication unit, target episodic replication unit, link, and schedule names.

Replicating compressed data

Using the `vxcompress` utility, episodic replication is able to replicate any compressed file that is created at the source to the target, while maintaining the same compression characteristics. The compression characteristics include the algorithm, the strength, and the block size. The data is read in the compressed format from the source, sent over the network, and written to the target system in the same format. This form of compression reduces the amount of storage that is required on the target system.

Note: Compressed files that are created using archive utilities such as .tar or .zip, are treated as normal data files and not compressed during episodic replication.

Displaying episodic replication job information and status

The `Replication> episodic job show` and `Replication> episodic job status` commands display job definition information, which allows you to confirm any changes that are made to your episodic replication job and view current job status.

The `Replication> episodic job show` command displays single job definition, or all of the job definitions for a destination cluster.

To display the job definitions

- ◆ To display the job definitions, enter the following command:

```
Replication> episodic job show [job_name]
```

job_name Enter the name of the job you want to display. If you want to list all of the job definitions, enter the command without a job name.

The `Replication> episodic job status` command displays the status of one or all of the jobs that are copied during episodic replication and the time the episodic replication occurred.

To display the status of an episodic replication job

- ◆ To display the status of an episodic replication job or all the jobs, enter the following command:

```
Replication> episodic job status job_name
```

job_name Enter the name of the job you want to display status for.

If a job is not specified, all status of all the jobs is displayed.

If the Job State displays `Trying_to_enable`, then the `job enable` command is in progress. Check the job status again after a few minutes.

Synchronizing an episodic replication job

To synchronize an enabled episodic replication job

- ◆ To synchronize an enabled episodic replication job, enter the following:

```
Replication> episodic job sync job_name
```

job_name Specify the name of the episodic replication job you want to synchronize.

Behavior of the file systems on the episodic replication destination target

Destination file systems are mounted as read-write. Read-only access is allowed, but you are not expected to modify the destination file system content. While episodic replication occurs, destination file systems may not be in a consistent state. To provide consistent images of the destination file systems at different stages of the episodic replication, the episodic replication service creates and manages Storage Checkpoints of each destination file system.

The episodic replication service creates a new destination Storage Checkpoint:

- Before the first session (before a full-sync)
- After every successful episodic replication session (after every incremental sync)

Storage Checkpoints are automatically mounted under the `.checkpoint` directory inside the target file system, for example:

```
/vx/target_mount/.checkpoint/ckpt_name
```

where *target_mount* is the name of the target file system and *ckpt_name* is the name of the Storage Checkpoint.

You can use the `Storage> snapshot list` command to view these Storage Checkpoints and you can use Veritas Access commands to export any of these Storage Checkpoints for read-only purposes. The episodic replication Storage Checkpoint names are prefixed with `vxfsrepl_` and also contain the Storage Checkpoint creation time.

Accessing file systems configured as episodic replication destinations

Destination Storage Checkpoints are automatically mounted and therefore cannot be brought online or taken offline using the `Storage> snapshot` commands. The destination Storage Checkpoints can only be accessed through the `.checkpoint` directory. This accessibility also applies to any user created Storage Checkpoints on the episodic replication destination file system.

Creating a recovery point objective (RPO) report

A recovery point is the last stable version of a set of data that can be used to recover from a site failure. A recovery point objective (RPO) is a goal set by an organization to define how much data an organization is willing to lose in the event of an outage. RPO is measured in terms of minutes (or hours) of data lost. For example, a company might have an RPO goal of two hours, which means no more than two hours of data should be lost because of a site failure.

Veritas Access episodic replication tracks the times that are associated with episodic replication jobs and enables you to generate an RPO report to help you manage your organization's RPO objectives. The RPO report shows the current RPO times for a episodic replication job. The report also shows the average RPO times over the last 24 hours, the last week, and the last month.

To create an RPO report

- ◆ To create an RPO report, enter the following:

```
Replication> episodic rpo show job_name [duration]
```

<i>job_name</i>	Enter the name of the episodic replication job that is associated with the RPO report.
<i>duration</i>	To show the average RPO over a specific number of hours, enter a numeric value to indicate the duration in hours. For example, enter 5 to show the average RPO values over the last five hours. The range of hours you can specify is 1 to 712.

Episodic replication job failover and failback

Typically, the source cluster drives a episodic replication session. However, in some situations, it may be useful for the destination cluster to drive the episodic replication session. Veritas Access supports a failover and a failback feature for episodic

replication jobs. This feature enables control of episodic replication jobs to be temporarily relocated from the source cluster to the destination (target) cluster.

Job failover and failback is useful for:

- **Planned failover**
In cases where the source cluster is taken down for routine maintenance or for moving applications to another cluster, a planned failover procedure is available for moving episodic replication jobs from the source cluster to the destination cluster.
- **Disaster recovery**
In cases where the source cluster fails unexpectedly, an unplanned failover procedure is available for moving episodic replication jobs to the destination cluster.

Note: In the event of a planned or unplanned failover from the source cluster to the destination cluster, there should be at least one successful sync attempt. The successful sync ensures that a consistent point in time image is present on the destination cluster that can be used for the failover.

With job failover and failback, you use the `Replication> episodic job failover` command to move control from the source cluster to the destination cluster. You use the `Replication> episodic job failback` to restore control to the source cluster. The `link_name` is the link of one of the destination clusters. The `link_name` argument can be empty when the source cluster is not available, in which case the job failover can be executed from one of the destination clusters.

Essentially, job failover takes job and episodic replication unit definitions from the episodic replication database on the source cluster and copies them to the episodic replication database on the destination cluster.

Warning: Job failover assumes that all episodic replication job names and episodic replication unit names are unique across all Veritas Access clusters on your network. Before you use the episodic replication failover feature, make sure that these names are unique.

After a job failover or failback, you must manually start or enable the episodic replication job to start pre-configured schedules. Link throttle information should be reconfigured after the job failover or failback.

Job failover does not automatically move the NFS or the CIFS share information that is associated with job failover episodic replication units from the source cluster to the destination cluster. Share information has to be done manually.

Table 26-1 Job failover and failback commands

Command	Definition
<code>episodic job failover</code>	Transfer control of an episodic replication job from the source cluster to the destination cluster.
<code>episodic job failback</code>	Return control of an episodic replication job from the destination cluster to the source cluster.

Process summary

The steps you take for job failover and failback vary depending the type of failover or failback you perform. Failover and failback types include:

- Planned failover
- Unplanned failover
- Failback after a planned failover
- Failback after an unplanned failover

Each process is summarized in the following sections. Typically, you would use the planned failover and planned failback processes in most situations.

Overview of the planned failover process

For planned failovers, most of the failover steps are executed from the source cluster.

- From the source cluster:
 - Stop all applications that access the replicated files. This step is recommended, but not required.
 - Use the `Replication> episodic job sync job_name` command to execute the job and make sure files on the source cluster and destination cluster are synchronized.
 - Use the `Replication> episodic job failover force=yes/no job_name current_cluster_link` command to move control of the job from the source cluster to the destination cluster.
- From the destination cluster:
 - Use the `Replication> episodic job enable job_name` command to enable the job or run a sync on the destination cluster.

- Use the `Replication> episodic job sync job_name` command to ensure that the episodic replication job is in a well-defined state and incremental episodic replication can be resumed.

Once the job is failed over, job control remains on the destination cluster until a planned failback is activated.

Overview of the planned failback process

After a job failover has been accomplished and the source cluster is ready to take back control of the episodic replication task, you can use the job failback feature to release control from the destination cluster and return it to the source cluster

- From the destination cluster:
 - Stop all applications that access the replicated files. This step is recommended, but not required.
 - Use the `Replication> episodic job sync job_name` command to execute the job and make sure files on the source cluster and destination cluster are synchronized.
 - Use the `Replication> episodic job disable job_name` command to disable the job.
- From the source cluster:
 - Use the `Replication> episodic job failback force=yes/no job_name current_cluster_link` command to move control of the job from the destination cluster back to the original source cluster.
 - Use the `Replication> episodic job enable job_name` command to enable the job or run a sync on the source cluster.
 - Use the `Replication> episodic job sync job_name` command to ensure that the episodic replication job is in a well-defined state and incremental episodic replication can be resumed.

Overview of the unplanned failover process

In some cases (for example, unexpected equipment failure), you may need to execute an unplanned failover for episodic replication jobs. The unplanned failover process differs from the planned failover process.

This section shows an overview of the steps you take to perform an unplanned failover.

For unplanned failovers, all the commands are executed from the destination cluster.

- Make sure that you are logged into the destination cluster.

- Use the `Replication> episodic job failover force=yes/no job_name` command to failover the job.

Overview of the unplanned fallback process

After an unplanned failover, when the source cluster comes up, you can use the following unplanned fallback process to return control to the original source cluster:

- Make sure that you are logged into the source cluster.

Note: Before starting the fallback process, verify that the episodic replication service is running on the master node. If the service is not running on the master node, stop the service using the `Replication> episodic service stop` command and start it again using the `Replication> episodic service start` command.

- Use the `Replication> episodic job failover force=yes/no job_name current_cluster_link` command to configure the current source cluster as a valid target to the new source cluster. This command should be executed from the old source cluster.
- Use the `Replication> episodic job sync job_name` command from the new source cluster to synchronize file system data with the newly added destination cluster.
- Use the `Replication> episodic job fallback force=yes/no job_name current_cluster_link` command to move control of the episodic replication job from the destination cluster back to the source cluster.
- Use the `Replication> episodic job sync job_name` command to ensure that the episodic replication job is in a well-defined state and incremental episodic replication can be resumed.

Note: An administrator can use the `Replication> episodic job destroy force` command to clean up local job configuration. Configuration of the other clusters, which are part of the job, will not be modified and any episodic replication units will be disassociated from job. The `Replication> episodic job destroy force` and `Replication> episodic repunit destroy force` commands should be used in the event of an unrecoverable configuration or episodic replication direction mismatch.

Configuring continuous replication

This chapter includes the following topics:

- [About Veritas Access continuous replication](#)
- [How Veritas Access continuous replication works](#)
- [Starting Veritas Access continuous replication](#)
- [Setting up communication between the source and the destination clusters](#)
- [Setting up the file system to replicate](#)
- [Managing continuous replication](#)
- [Displaying continuous replication information and status](#)
- [Unconfiguring continuous replication](#)
- [Continuous replication failover and failback](#)

About Veritas Access continuous replication

The Veritas Access continuous replication solution provides high performance, robustness, ease of use, and synchronous replication capability which is designed to contribute to an effective disaster recovery plan.

Veritas Access continuous replication lets you replicate volumes from one node in the source cluster to another node in the destination cluster. The continuous replication enables you to maintain a consistent copy of application data at one remote location. It replicates the application writes on the volumes at the source location to a remote location across any distance.

If a disaster occurs at the source location, you can use the copy of the application data at the remote location and restart the application at the remote location. The host at the source location on which the application is running is known as the primary host and the host at the target location is known as the secondary host. The volumes on the primary host must be synchronized initially with the volumes on the secondary host.

Major features of Veritas Access continuous replication include:

- Performs replication of volumes in synchronous as well as asynchronous mode, ensuring data integrity and consistency.
- Replicates data for cluster file system as well as a scale-out file system
- Maintains write-order fidelity, which applies writes on the secondary host in the same order that they were issued on the primary host.
- Enables easy recovery of the application at the remote site.
- Provides both command-line interface (CLISH) and the graphical user interface (GUI) for online management of the synchronous replication.

See the `continuous(1)` manual page for more information.

How Veritas Access continuous replication works

The Veritas Access continuous replication implements Cluster Volume Replication (CVR) method internally. It is based on volume level replication.

It includes the following components:

- **Replicated Volume Group (RVG)**
A Replicated Volume Group (RVG) is a group of volumes or file sets within a given Volume Manager (VxVM) disk group which is configured for replication. The volumes in an RVG are consistent while replicating data. An RVG is always a subset of a VxVM disk group. One or more related volumes in a disk group can be configured as an RVG.
Volumes that are associated with an RVG and contain application data are called data volumes. The data volumes in the RVG are under the control of an application.
During replication, the write-order is strictly maintained within an RVG to ensure that each remote volume is always consistent, both internally and with all other volumes of the group. CVR replicates data from a primary RVG on the host (where the application is running) to the secondary RVG. An RVG also contains the Storage Replicator Log (SRL) and Replication Link (RLINK), which are used internally by CVR.
- **Storage Replicator Log (SRL)**

The Storage Replicator Log (SRL) is a circular buffer of writes for an RVG. Each RVG contains one SRL. Writes to the data volumes in the RVG are first queued in the SRL on the primary host before they are sent to the secondary. CVR uses the SRL to track the order of writes to data volumes in the RVG. The SRL enables continuous replication to maintain write-order fidelity at the secondary RVG.

- **Replication Link (RLINK)**

An RLINK is associated with an RVG and establishes the link between the primary and the secondary RVG. Each RLINK associated with a primary RVG represents one secondary. Each RLINK associated with a secondary RVG represents the primary.

- **Data Change Map (DCM)**

The Data Change Map (DCM) is another important component of CVR that is used to track writes when the SRL overflows. It enables you to avoid complete resynchronization of the data on the secondary. The DCM contains a bitmap. The DCM is active only on the primary side. The DCM becomes active only when the SRL is no longer large enough to hold accumulated updates. While the DCM is active, each bit that has been set in the DCM represents a region whose contents are different between the primary and the secondary.

- **Replicated Data Set (RDS)**

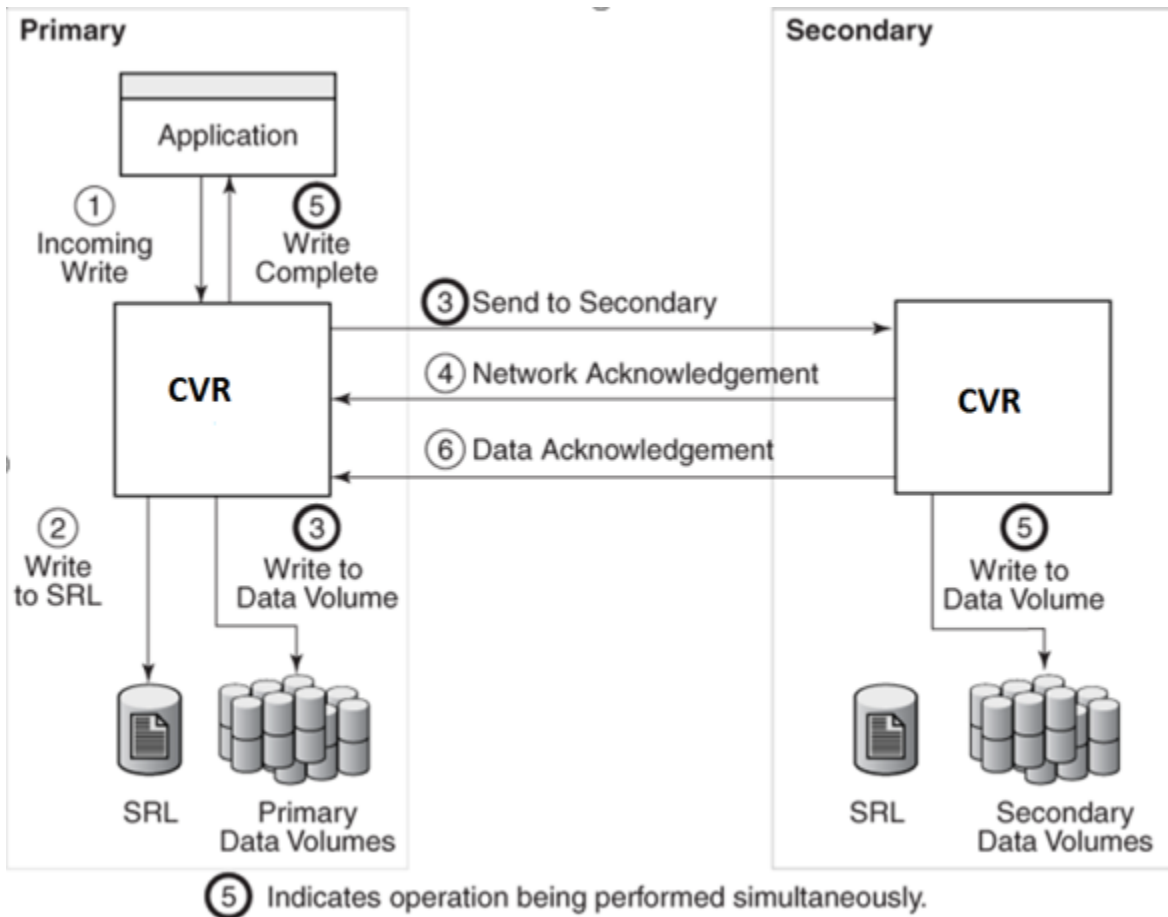
A Replicated Volume Group (RVG) on the primary host and its counterparts on the secondary hosts make up a Replicated Data Set (RDS). An RDS enables grouping of the RVG on the primary and its counterparts on the secondary. The concept of primary host and secondary host is used only in the context of a particular Replicated Data Set (RDS).

Note: The cloud tier data does not get replicated to a new bucket. The same cloud bucket is accessed from the destination cluster.

How data flows in continuous replication synchronous mode

In synchronous mode of continuous replication, CVR processes an incoming write by performing the following steps.

Figure 27-1 Data flow in continuous replication synchronous mode



1. CVR receives a write on the primary host.
2. CVR writes it to the primary SRL.
3. CVR sends the write to the secondary host and waits for the synchronous network acknowledgments from the secondary hosts. At the same time, CVR writes to the data volumes on the primary host.
4. On the secondary host, CVR receives the write, processes it, and sends a network acknowledgment to the primary host.
5. CVR sends writes to the data volumes on the secondary host. When the primary host receives a network acknowledgment from the secondary host, CVR acknowledges to the application that the write is complete.

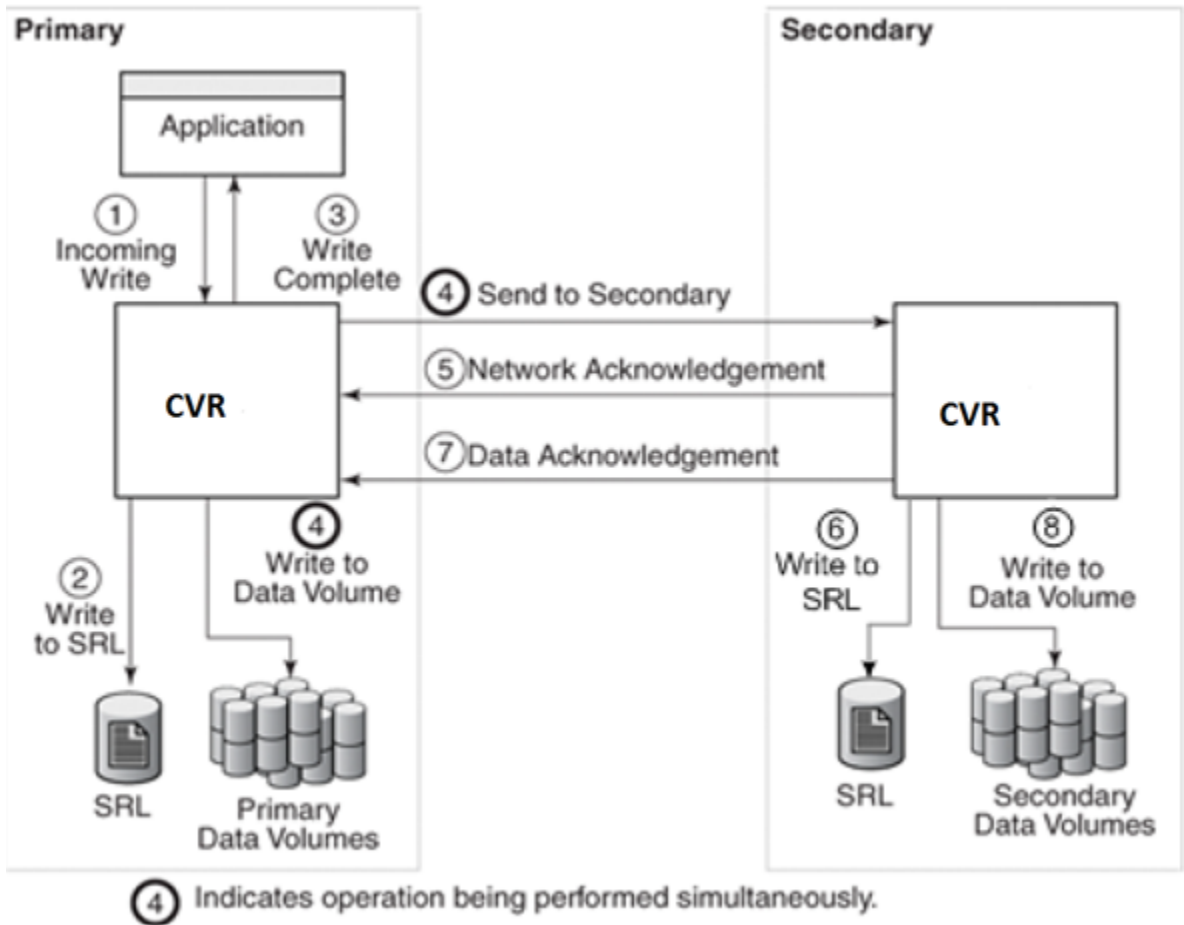
Note: The secondary RVG sends the network acknowledgment as soon as the write is received in the CVR kernel memory. This removes the time required to write to the secondary data volumes from the application latency. On the primary host, CVR does not wait for data to be written to the secondary data volumes. This improves application performance. However, CVR tracks all such acknowledged writes that have not been written to the data volumes. CVR can replay these tracked writes if the secondary host crashes before writing to the data volumes on the secondary host or if the primary host crashes before it receives the data acknowledgment.

When the write is written to the data volumes on the secondary host, CVR on the secondary host sends a data acknowledgment to the primary host. CVR marks the write as complete in the SRL when the primary receives the data acknowledgment from all the secondary hosts.

How data flows in continuous replication asynchronous mode

In asynchronous mode of continuous replication, CVR processes an incoming write by performing the following steps.

Figure 27-2 Data flow in continuous replication asynchronous mode



1. CVR receives a write on the primary host.
2. CVR writes it to the primary SRL.
3. On the primary host, CVR acknowledges to the application that the write is complete.
4. CVR sends the writes to the asynchronous secondary host in the order in which they were received on the primary host. At the same time, CVR writes to the primary data volumes.
5. When the primary host receives the network acknowledgment, it knows that the write has been received in the secondary CVR memory buffer.

6. CVR sends the writes to the data volumes on the secondary host and then sends a data acknowledgment to the primary host.
7. When the primary host receives the data acknowledgment, CVR marks the write as complete in the SRL.

Starting Veritas Access continuous replication

This section lists the specific commands that are needed to run Veritas Access continuous replication on your clusters.

Ensure the following before starting replication:

- Before you set up your clusters for continuous replication, you must first identify which is the source cluster and which is the destination cluster. All of the commands are performed on the source cluster first.
- Make sure both the source cluster and the destination cluster have the same version of Veritas Access.
- To use Veritas Access continuous replication, you have to first create an online file system on the Veritas Access source cluster.
- Assign a virtual IP (VIP) address to both the source and the destination clusters. The Veritas Access continuous replication service requires VIP addresses not already in use for the two clusters to communicate.

To start Veritas Access continuous replication on the source cluster

- 1 To bind a virtual IP address for the continuous replication service on the source cluster, enter the following:

```
Replication> continuous config bind ip_addr [device] [netmask]
```

ip_addr Virtual IP address for the continuous replication service on the destination cluster.

device The public network interface name that you want the replication IP address to use.

netmask Netmask for the replication IP address.

- 2 To start the continuous replication service, enter the following on the source node:

```
Replication> continuous service start [nodename]
```

nodename The name of the node in the local cluster where you want to start the replication service.

- 3 To check the status of the continuous replication service, enter the following:

```
Replication> continuous service status
```

- 4 To confirm the IP address is up and running, enter the following:

```
Replication> continuous config show ip
```

Note: Alternately, you can use the `Network> ip addr show` command to confirm that the IP address is up and running.

To start Veritas Access continuous replication on the destination cluster

- 1 To bind a virtual IP address for the replication service on the destination cluster, enter the following:

```
Replication> continuous config bind ip_addr [device] [netmask]
```

ip_addr Virtual IP address for the continuous replication service on the destination cluster.

device The public network interface name that you want the replication IP address to use.

netmask Netmask for the replication IP address.

- 2 To start the continuous replication service, enter the following on the destination node:

```
Replication> continuous service start [nodename]
```

nodename The name of the node in the local cluster where you want to start the replication service.

- 3 To check the status of the continuous replication service, enter the following:

```
Replication> continuous service status
```

- 4 To confirm that the IP address is up and running, enter the following:

```
Replication> continuous config show ip
```

You next need to set up communication between the source and the destination clusters.

Setting up communication between the source and the destination clusters

You need to set up communication between your source and your destination clusters.

Make sure that you already created an online file system on the Veritas Access source cluster and an online file system on the Veritas Access destination cluster.

Veritas Access Replication authentication strategy is based on RSA-key authentication, and both the source and the destination clusters have to export their

replication public keys. The source cluster imports the destination cluster's public key and the destination cluster imports the source cluster's public key.

After you have determined which two Veritas Access clusters to use, you need to authenticate them.

The `Replication> continuous config` commands must be executed in a specific order.

- You need to run the `Replication> continuous config bind` command (to bind the virtual IP) before you can run the `Replication> continuous service start` command.
- You need to run the `Replication> continuous config export_keys` and `Replication> continuous config import_keys` to export and import the keys of both the source and the destination clusters.
- You can only run the `Replication> continuous config auth` command after both the source and destination have imported each other's keys.
- You need to run the `Replication> continuous config auth` command to create a link from every cluster to any remaining cluster that is used for replication irrespective of their role as a source or a destination cluster.

This command checks the two-way communication between the source and the destination cluster, and authenticates the clusters allowing the Veritas Access continuous replication service to begin.

This section provides a walk-through for the creation and export/import of these encrypted keys for both the source and the destination cluster.

Note: Without the correct authentication of the source and the destination encryption keys, Veritas Access continuous replication does not function correctly.

To export the source cluster's key to the destination cluster

- 1 To export the source cluster's key to the destination cluster, enter the following:

```
Replication> continuous config export_keys [URL]
```

URL The location you want to copy the public keys to.

If you do not want to enter a URL, you can copy the output from the `Replication> continuous config export_keys` command into the `Replication> continuous config import_keys` command at the destination cluster.

By default, the output is displayed on your computer screen.

The SCP and FTP protocols are supported.

- 2 To import the source cluster's key to the destination cluster, enter the following:

```
Replication> continuous config import_keys [URL/keyfile]
```

URL The location you want to copy the public keys from.

keyfile The file name of the key that is generated by the export.

If you did not enter a URL during the `Replication> continuous config export_keys` command, you can cut and paste the output and enter it into the `Replication> continuous config import_keys` command.

- 3 To verify that the key has been imported correctly, enter the following:

```
Replication> continuous config show
```

To export the destination cluster's key to the source cluster

- 1 To export the destination cluster's key to the source cluster, enter the following:

```
Replication> continuous config export_keys [URL]
```

URL The location you want to copy the public keys to.

The SCP and FTP protocols are supported.

If you do not want to enter a URL, you can cut and paste the output from the
 Replication> continuous config export_keys command to the
 Replication> continuous config import_keys command. By default, the
 output is displayed to your computer screen.

- 2 To import the destination cluster's key to the source cluster, enter the following:

```
Replication> continuous config import_keys [URL/keyfile]
```

URL Enter the URL of the location you want to copy the public keys from.

keyfile Enter the file name of the key that is generated by the export.

If you did not enter a URL during the Replication> continuous config
 export_keys command, you can cut and paste the output and enter it into the
 Replication> continuous config import_keys command.

- 3 To verify that the key has been imported correctly, enter the following:

```
Replication> continuous config show
```

To authenticate source cluster and destination clusters for replication

- 1 This command should be executed on the source cluster as well as on the destination cluster. To authenticate the public keys on the source cluster and the destination clusters, enter the following:

```
Replication> continuous config auth conIP link_name
```

conIP Enter the destination cluster console IP address.

link_name Both the source cluster and the destination cluster need to be assigned a unique identifier (name). This identifier is used to identify the link that is established between the source and the destination clusters. You can use the link name instead of the virtual IP addresses of the source and the destination clusters when using the other replication commands. For example:
Pune_Shanghai.

- 2 To confirm the authentication, enter the following:

```
Replication> continuous config show
```

Note: These steps must be executed on the destination side cluster to authenticate the public keys on the source and the destination cluster.

Once you have configured the clusters and links, you need to set up the file system you want to replicate.

Setting up the file system to replicate

You need to set up the file systems you want to replicate. The file system which you are going to replicate should be in online state. The continuous replication is supported for cluster file system as well as scale out file system.

To set up a continuous replication, enter the following:

```
Replication> continuous enable fs_name pool_name link_name  
[delayed=yes/delayed=no]
```

fs_name The name of the file system that you want to replicate from source to destination cluster.

pool_name The name of the pool.

<i>link_name</i>	The link name which was created during authentication time.
<i>delayed=yes</i> <i>/delayed=no</i>	The delayed parameter value is yes if you want to set up continuous asynchronous replication. By default, its value is no .

The command configures the continuous replication between the source and the destination cluster. A file system is created with the same name and same size on the destination cluster. The file system is in offline state at the secondary site to maintain a consistent copy of data.

It requires the pool on the destination cluster to have sufficient storage to create a file system and Storage Replicated Log (SRL) volume. The pool name should be same at the source and the destination cluster. The SRL volume size is one-fourth of the file system size.

For example, if the file system size on the source cluster is 8 GB then at least 3 GB storage should be present in the pool at the source cluster to create an SRL volume. For the target cluster, 11 GB storage should be present in the pool to create the file system and SRL volume.

Note: Continuous replication is not supported for file system with encoded layout and encrypted volume. When setting up replication, Veritas recommends that you do not make any modifications or deletions on the target side of the file system. File system grow and shrink operations are not supported on file systems which are configured under continuous replication. The mode of replication (synchronous or asynchronous) cannot be changed after configuring the replication.

Managing continuous replication

You can manage continuous replication by `Replication> continuous start`, `Replication> continuous stop`, `Replication> continuous pause` and `Replication> continuous resume` commands. It is recommended that you should execute all these commands from the source cluster.

To start data replication

- ◆ To start data replication, enter the following command:

```
Replication> continuous start fs_name
```

fs_name Specify the file system name that you have configured for continuous replication.

The data replication between source cluster and destination cluster starts.

To pause data replication

- ◆ To pause data replication, enter the following command:

```
Replication> continuous pause fs_name
```

fs_name Specify the file system name that you have configured for continuous replication.

The data replication between source cluster and destination cluster is paused.

To resume data replication

- ◆ To resume data replication, enter the following command:

```
Replication> continuous resume fs_name
```

fs_name Specify the file system name that you have configured for continuous replication.

The data replication between source cluster and destination cluster which was paused is resumed.

To stop data replication

- ◆ To stop data replication, enter the following command:

```
Replication> continuous stop fs_name
```

fs_name Specify the file system name that you have configured for continuous replication.

The data replication between source cluster and destination cluster is stopped.

Note: Veritas recommends that you should execute `Replication> continuous resume` command when replication is in paused state. You should execute `Replication> continuous stop` command when replication is in progress.

Displaying continuous replication information and status

The `Replication> continuous show` and `Replication> continuous status` commands display information on continuous replication which allows you to confirm any changes that are made to your replication file system and view the current file system status.

To display the list of file systems which are configured under continuous replication

- ◆ To display the list of file systems which are configured under continuous replication, enter the following command:

```
Replication> continuous show
```

To display the status of a replication file system

- ◆ To display the status of a replication file system, enter the following command:

```
Replication> continuous status fs_name
```

fs_name Specify the file system name that you have configured for continuous replication.

Table 27-1 describes the important attributes displayed by the `Replication> continuous status` command.

Table 27-1

Attribute	Description
Replicated Data Set	Specifies the name of the replicated data set
Replication role	Specifies the role of the cluster in continuous replication. It is either primary or secondary.
Replication link	Specifies the link name which is created during the authentication of the source and the destination cluster.

Table 27-1 (continued)

Attribute	Description
Primary site Info	Provides the details of continuous replication related to the source cluster like host name and RVG state.
Secondary site Info	Provides the details of continuous replication related to the destination cluster like host name, configured mode, data status, replication status, current mode and timestamp information.
Host name	<p>For the primary site, it provides the continuous replication IP which binds on the source cluster using the <code>Replication> continuous config bind</code> command.</p> <p>For the secondary site, it provides the continuous replication IP which binds on the destination cluster using the <code>Replication> continuous config bind</code> command.</p>
RVG state	<p>Specifies the state of the primary RVG.</p> <p>See Table 27-2 for details on its various states.</p>
Configured mode	Specifies the continuous replication configured mode. It may be <i>synchronous-override</i> or <i>asynchronous</i> .
Current mode	Specifies the mode of replication - <i>asynchronous</i> or <i>synchronous</i> , that is used to replicate data to the secondary.
Data status	<p>Shows the data status for the secondary.</p> <p>See Table 27-3 for details on its various states.</p>
Replication status	<p>Specifies the status of the replication to the secondary.</p> <p>See Table 27-4 for details on its various states.</p>

Table 27-1 (continued)

Attribute	Description
Logging to	Indicates whether updates for the secondary are tracked on the primary using the SRL or DCM. See Table 27-5 for details on its various states.
Timestamp information	Shows the time by which secondary is lagging behind the primary.

[Table 27-2](#) describes the values for the RVG state.

Table 27-2 RVG status

Value	Description
acting_secondary	The primary RVG is currently the acting secondary as part of the fast failback process. Writes to the data volumes in this RVG are disabled independent of whether the RVG is started or stopped.
disabled for I/O	Primary RVG is disabled for I/O. The RVG is stopped.
enabled for I/O	Primary RVG is enabled for I/O. The RVG has been started.
needs recovery	State of the RVG after an import or reboot.
Passthru	The primary RVG is in passthru mode because the primary SRL is detached or missing.

[Table 27-3](#) describes the values for the Data status.

Table 27-3 Data status

Value	Description
consistent, behind	Secondary data is consistent but not up-to-date with the primary data.
consistent, stale	The data on the secondary is consistent. Replication to the secondary has been stopped. The primary RLINK is detached.

Table 27-3 Data status (*continued*)

Value	Description
consistent, up-to-date	The secondary data is consistent and is current or up-to-date with the primary data. The primary role can be migrated to the secondary.
inconsistent	The data on the secondary volumes is not consistent and the secondary cannot take over.
N/A	Current state of the secondary data cannot be determined. This may occur because of a configuration error on the secondary.

[Table 27-4](#) describes the values for the Replication status.

Table 27-4 Replication status

Value	Description
logging to DCM	DCM is active for the secondary. New updates on primary are tracked using DCM for the secondary. The following information may be displayed: <ul style="list-style-type: none"> ■ <i>needs dcm resynchronization</i>—Resynchronize the secondary using DCM resynchronization to continue replication.
needs failback synchronization	The primary RVG is acting as secondary as part of the fast failback process. Start failback resynchronization on the new primary to continue replication.
not replicating	Data is not being replicated to secondary because primary RLINK is in needs_recovery state. <ul style="list-style-type: none"> ■ <i>primary needs_recovery</i>—Primary RLINK needs to be recovered before replication can resume.

Table 27-4 Replication status (*continued*)

Value	Description
paused by user	Replication to secondary is paused due to some administrative action. This results in the following states: <ul style="list-style-type: none"> ■ <i>primary paused</i>—primary RLINK is paused. ■ <i>secondary paused</i>—secondary RLINK is paused.
paused due to error	Replication to secondary is paused due to the following errors: <ul style="list-style-type: none"> ■ <i>secondary config error</i>—secondary has some configuration error. ■ <i>secondary log error</i>—secondary SRL has an I/O error.
paused due to network disconnection	Replication to secondary is paused due to some network problem.
replicating	Replication can take place if there are updates on the primary data volumes.
resync in progress	Resynchronization to the secondary is in progress. <ul style="list-style-type: none"> ■ <i>autosync</i>—Resynchronization type is autosync. ■ <i>dcm resynchronization</i>—resynchronization after an SRL overflow. ■ <i>failback resynchronization</i>—resynchronization using failback logging. ■ <i>smartsync</i>—resynchronization type is autosync using SmartMove.
resync paused by user	Resynchronization to secondary is paused due to some administrative action. This results in the following states: <ul style="list-style-type: none"> ■ <i>primary paused</i>—primary RLINK is paused. ■ <i>secondary paused</i>—secondary RLINK is paused.

Table 27-4 Replication status (*continued*)

Value	Description
resync paused due to error	Resynchronization to Secondary is paused because of the following errors: <ul style="list-style-type: none"> ■ <i>secondary config error</i>—secondary has some configuration error. ■ <i>secondary log error</i>—Secondary SRL has an I/O error.
resync paused due to network disconnection	Resynchronization to secondary is paused due to some network problem.
stopped	Replication to secondary is stopped due to the following: <ul style="list-style-type: none"> ■ <i>Primary detached</i>—Primary RLINK is detached ■ <i>Secondary detached</i>—Secondary RLINK is detached.
N/A	The replication status cannot be determined.

Table 27-5 describes the values for the Logging to field.

Table 27-5 Logging to status

Value	Description
DCM (contains xxx Kbytes) (log_type)	DCM is active (in use) for the replication to the secondary. The <i>log_type</i> can be <i>autosync</i> , <i>failback logging</i> , or <i>SRL protection logging</i> . The yyy% value can sometimes reach beyond 100%. If synchronization is restarted and the DCM map is full, new incoming writes cause the total yyy% to exceed 100%.
SRL (xxx Kbytes behind, yyy % full)	Updates to be transferred to secondary are logged into the SRL and are currently occupying xxx Kbytes or yyy% of the SRL.
SRL	SRL is used for logging. Check the Data status field for the status of the secondary data.

Unconfiguring continuous replication

You can unconfigure continuous replication.

To unconfigure continuous replication

- 1 Stop the replication. Before you disable continuous replication, you have to stop replication using the following command.

```
Replication> continuous stop <fs_name>
```

fs_name Specify the file system.

Note: This command should be executed from the source cluster.

- 2 Check the replication status.

```
Replication> continuous status <fs_name>
```

fs_name Specify the file system name.

- 3 Disable continuous replication. All the configuration which was done for replication configuration when you enabled continuous replication from the source and the destination cluster is destroyed.

```
Replication> continuous disable <fs_name> <link_name>
```

fs_name Specify the file system name.

link_name Specify the link name.

Note: This command should be executed from the source cluster.

4 Delete the authentication links.

```
Replication> continuous config deauth <link_name>
```

link_name Specify the link name.

Note: The `Replication> continuous config deauth` command should be executed from the source cluster. The command deletes the link from the destination to the source cluster.

5 Delete the keys from the source and the destination clusters.

```
Replication> continuous config del_keys <remote_console_ip>
```

remote_console_ip Specify the remote console IP address.

6 Stop the continuous service from the source and the destination clusters.

```
Replication> continuous service stop
```

7 Unbind the replication IP from the source and the destination clusters.

```
Replication> continuous config unbind <replication_ip>
```

replication_ip Specify the replication IP address.

Continuous replication failover and failback

Typically, the source cluster drives a replication session. However, in some situations, it may be useful for the destination cluster to drive the replication session. Veritas Access supports a failover and a failback feature for continuous replication. This feature enables control of replication to be temporarily relocated from the source cluster to the destination (target) cluster.

Continuous replication failover and failback is useful for:

- **Planned failover**
 In cases where the source cluster is taken down for routine maintenance or for moving applications to another cluster, a planned failover procedure is available for moving replication from the source cluster to the destination cluster.
- **Disaster recovery**

In cases where the source cluster fails unexpectedly, an unplanned failover procedure is available for moving replication to the destination cluster.

With failover and failback, you can use the `Replication> continuous failover` command to move control from the source cluster to the destination cluster. You use the `Replication> continuous failback` to restore control to the source cluster.

Continuous replication failover does not automatically move the NFS or the CIFS share information that is associated with file system from the source cluster to the destination cluster. Share information has to be done manually.

Process summary

The steps you take for failover and failback vary depending the type of failover or failback you perform. Failover and failback types include:

- Planned failover
- Unplanned failover
- Failback after a planned failover
- Failback after an unplanned failover

Each process is summarized in the following sections. Typically, you would use the planned failover and planned failback processes in most situations.

Overview of the planned failover process

For planned failover, execute the following command:

```
Replication> continuous failover fs_name
```

Where *fs_name* is the name of the file system which is configured under continuous replication.

Once a planned failover happens, the roles of primary and secondary are switched. It will online the file system at new primary site and offline the file system at new secondary site.

Note: Planned failover command should be run when both the source and the destination clusters are reachable from each other. It should be executed from the source cluster and replication should be in progress. If you have NFS/CIFS shares on the source cluster, it is recommended that you should stop the NFS/CIFS server before planned failover.

Overview of the planned failback process

After a planned failover has been accomplished and the source cluster is ready to take back control of the replication task, you can use the failback feature to release control from the destination cluster and return it to the source cluster.

For planned failback, execute the following command:

```
Replication> continuous failback fs_name
```

Where *fs_name* is the name of the file system which is configured under continuous replication.

Note: Planned failback command should be run when both the source and the destination clusters are reachable from each other. It should be executed from the destination cluster and replication should be in progress. If you have NFS/CIFS shares on the source cluster, it is recommended that you should stop the NFS/CIFS server before planned failback.

Overview of the unplanned failover process

In some cases (for example, unexpected equipment failure), you may need to execute an unplanned failover for replication. The unplanned failover process differs from the planned failover process.

For unplanned failover, execute the following command from the destination cluster:

```
Replication> continuous failover fs_name
```

Where *fs_name* is the name of the file system which is configured under continuous replication.

Once an unplanned failover happens, the destination cluster becomes the new source cluster. It will online the file system at the new source cluster.

Note: Though the commands used for planned and unplanned failover are the same, the intention and pre-requisites are different. For unplanned failover, the source cluster should be unreachable from the destination cluster.

Overview of the unplanned failback process

After an unplanned failover, when the source cluster comes up, you can use the following unplanned failback process to retain the replication between source and destination cluster.

When the source cluster comes up, it still acts as the primary cluster. At this time, both the source and the destination clusters show the same status.

For unplanned failback, execute the following command from the original source cluster:

```
Replication> continuous failback fs_name
```

Where *fs_name* is the name of the file system which is configured under continuous replication.

Once, an unplanned failback happens, the original source cluster becomes the new destination cluster.

Note: Unplanned failback command should be run from the original source cluster. If you have NFS/CIFS shares on the original source cluster, it is recommended that you stop the NFS/CIFS server before unplanned failback.

Using snapshots

This chapter includes the following topics:

- [About snapshots](#)
- [Creating snapshots](#)
- [Displaying snapshots](#)
- [Managing disk space used by snapshots](#)
- [Bringing snapshots online or taking snapshots offline](#)
- [Restoring a snapshot](#)
- [About snapshot schedules](#)
- [Configuring snapshot schedules](#)
- [Managing automated snapshots](#)

About snapshots

A snapshot is a virtual image of the entire file system. You can create snapshots of a parent file system on demand. Physically, it contains only data that corresponds to the changes that are made in the parent, and so consumes significantly less space than a detachable full mirror.

Note: You cannot create a snapshot of a scale-out file system.

See [“About scale-out file systems”](#) on page 211.

Snapshots are used to recover from data corruption. If files, or an entire file system, are deleted or become corrupted, you can replace them from the latest uncorrupted snapshot. You can mount a snapshot and export it as if it were a complete file

removable

Valid values are:

- yes
- no

If the removable attribute is `yes`, the snapshot is removed automatically if the file system runs out of space.

The default value is `removable=no`.

Displaying snapshots

You can display all snapshots, or the snapshots taken of a specific file system or specific schedule of a file system. The output displays the snapshot name and the properties of the snapshots such as creation time and size.

To display snapshots

- ◆ To display snapshots, enter the following:

```
Storage> snapshot list [fs_name] [schedule_name]
```

fs_name Displays all of the snapshots of the specified file system. If you do not specify a file system, snapshots of all of the file systems are displayed.

schedule_name Displays the schedule name. If you do not specify a schedule name, then snapshots created under *fs_name* are displayed.

```
Storage> snapshot list
```

Snapshot	FS	Status
=====	==	=====
snap2	fs1	offline
sc1_24_Jul_2009_21_34_01_IST	fs1	offline
sc1_24_Jul_2009_19_34_02_IST	fs1	offline
presnap_sc1_24_Jul_2009_18_34_02_IST	fs1	offline
sc1_24_Jul_2009_17_34_02_IST	fs1	offline

ctime	mtime	Removable	Preserved	Size
=====	=====	=====	=====	=====
2009.Jul.27.02:40:43	2009.Jul.27.02:40:57	no	No	190.0M
2009.Jul.24.21:34:03	2009.Jul.24.21:34:03	yes	No	900.0M
2009.Jul.24.19:34:04	2009.Jul.24.19:34:04	yes	No	7.0G
2009.Jul.24.18:34:04	2009.Jul.24.18:34:04	yes	Yes	125M
2009.Jul.24.17:34:04	2009.Jul.24.17:34:04	yes	No	0K

Snapshot	Displays the name of the created snapshots.
FS	Displays the file systems that correspond to each created snapshots.
Status	Displays whether or not the snapshot is mounted (that is, online or offline).
ctime	Displays the time the snapshot was created.
mtime	Displays the time the snapshot was modified.
Removable	Determines if the snapshot should be automatically removed in case the underlying file system runs out of space. You entered either yes or no in the <code>snapshot create snapshot_name fs_name [removable]</code>
Preserved	Determines if the snapshot is preserved when all of the automated snapshots are destroyed.
Size	Displays the size of the snapshot.

Managing disk space used by snapshots

To manage the disk space used by snapshots, you can set a snapshot quota or capacity limit for the file system. When all of the snapshots for the file system exceed the capacity limit, snapshot creation is disabled for the file system.

You can also remove unnecessary snapshots to conserve disk space.

To enable snapshot quotas

- 1 To display snapshot quotas, enter the following:

```
Storage> snapshot quota list
FS           Quota      Capacity Limit
==          =====
fs1          on           1G
fs2          off          0
fs3          off          0
```

- 2 To enable a snapshot quota, enter the following:

```
Storage> snapshot quota on fs_name [capacity_limit]
```

fs_name	Specifies the name of the file system.
capacity_limit	Specifies the number of blocks used by all the snapshots for the file system. Enter a number followed by K, M, G, or T (for kilo, mega, giga, or terabyte). The default value is 0.

- 3 If necessary, you can disable snapshot quotas. You can retain the value of the capacity limit. To disable a snapshot quota, enter the following:

```
Storage> snapshot quota off [fs_name] [remove_limit]
```

fs_name	Specifies the name of the file system.
remove_limit	Specifies whether to remove the capacity limit when you disable the quota. The default value is true, which means that the quota capacity limit is removed. The value of false indicates that the quota is disabled but the value of the capacity limit remains unchanged for the file system.

To destroy a snapshot

- ◆ To destroy a snapshot, enter the following:

```
Storage> snapshot destroy snapshot_name
                        fs_name
```

snapshot_name Specifies the name of the snapshot to be destroyed.

fs_name Specifies the name of the file system from which the snapshot was taken. Snapshots with the same name could exist for more than one file system. In this case, you must specify the file system name.

Bringing snapshots online or taking snapshots offline

If you want to mount a snapshot through NFS or export a CIFS snapshot, you must bring the snapshot online. You can then create a CIFS or an NFS share using the snapshot name as the path. For example: /vx/fs1:snap1. The snapshot can only be mounted through NFS or exported through CIFS if it is online.

To bring a snapshot online

- ◆ To bring a snapshot online:

```
Storage> snapshot online snapshot_name fs_name
```

snapshot_name Specifies the name of the snapshot.

fs_name Specifies the name of the file system from which the snapshot was taken. Snapshots with the same name could exist for more than one file system. In this case, you must specify the file system name.

To take a snapshot offline

- ◆ To take a snapshot offline:

```
Storage> snapshot offline snapshot_name fs_name
```

snapshot_name Specifies the name of the snapshot.

fs_name Specifies the name of the file system from which the snapshot was taken. Snapshots with the same name could exist for more than one file system. In this case, you must specify the file system name.

Restoring a snapshot

This operation restores the file system to the state that is stored in the specified snapshot. When you restore the file system to a particular snapshot, snapshots taken after that point in time are no longer relevant. The restore operation also deletes these snapshots.

The restore snapshot operation prompts you for confirmation. Be sure that you want to restore the snapshot before responding yes.

To restore a snapshot

- ◆ To restore a snapshot, enter the following:

```
Storage> snapshot restore snapshot_name  
fs_name
```

snapshot_name Specifies the name of the snapshot to be restored.

fs_name Specifies the name of the file system to be restored.

About snapshot schedules

The `Storage> snapshot schedule` commands let you automatically create or remove snapshots for a file system at a specified time. The schedule indicates the time for the snapshot operation as values for minutes, hour, day-of-the-month, month, and day-of-the-week. The schedule stores these values in the crontab along with the name of the file system.

For example, `snapshot schedule create schedule1 fs1 30 2 * * *` automatically creates a snapshot every day at 2:30 AM, and does not create

snapshots every two and a half hours. If you wanted to create a snapshot every two and a half hours with at most 50 snapshots per schedule name, then run `snapshot schedule create schedule1 fs1 50 */30 */2 * * *`, where the value `*/2` implies that the schedule runs every two hours. You can also specify a step value for the other parameters, such as day-of-month or month and day-of-week as well, and you can use a range along with a step value. Specifying a range in addition to the `numeric_value` implies the number of times the crontab skips for a given parameter.

Automated snapshots are named with the schedule name and a time stamp corresponding to their time of creation. For example, if a snapshot is created using the name `schedule1` on February 27, 2016 at 11:00 AM, the name is:

```
schedule1_Feb_27_2016_11_00_01_IST.
```

Note: If the master node is being rebooted, snapshot schedules will be missed if scheduled during the reboot of the master node.

Configuring snapshot schedules

You can use snapshot schedules to automate creation of snapshots at regular intervals. The snapshot limit defines how many snapshots to keep for each schedule.

In some instances, snapshots may skip scheduled runs.

This may happen because of the following:

- When a scheduled snapshot is set to trigger, the snapshot needs to gain a lock to begin the operation. If any command is issued from the CLI or is running through schedules, and if the command holds a lock, the triggered snapshot schedule is not able to obtain the lock, and the scheduled snapshot fails.
- When a scheduled snapshot is set to trigger, the snapshot checks if there is any instance of a snapshot creation process running. If there is a snapshot creation process running, the scheduled snapshot aborts, and a snapshot is not created.

To create a snapshot schedule

- ◆ To create a snapshot schedule, enter the following:

```
Storage> snapshot schedule create schedule_name
      fs_name
      max_snapshot_limit
      minute [hour] [day_of_the_month]
      [month] [day_of_the_week]
```

For example, to create a schedule for an automated snapshot creation of a given file system at 3:00 am every day, enter the following:

```
Storage> snapshot schedule create schedule1 fs1 100 0 3 * * *
```

When an automated snapshot is created, the entire date value is appended, including the time zone.

schedule_name	<p>Specifies the name of the schedule corresponding to the automatically created snapshot.</p> <p>The <i>schedule_name</i> cannot contain an underscore ('_') as part of its value. For example, <i>sch_1</i> is not allowed.</p>
fs_name	<p>Specifies the name of the file system. The file system name should be a string.</p>
max_snapshot_limit	<p>Specifies the number of snapshots that can be created for a given file system and schedule name. The value is a numeric value between 1-366.</p> <p>When the number of snapshots reaches the limit, then the oldest snapshot is destroyed. If you decrease the limit for an existing schedule, then multiple snapshots may be destroyed (oldest first) until the number of snapshots is less than the maximum snapshot limit value.</p> <p>Note: If you need to save daily snapshots for up to one year, the <i>max_snapshot_limit</i> is 366.</p>
minute	<p>This parameter may contain either an asterisk like <i>*/15</i>", which implies every 15 minutes, or a numeric value between 0-59.</p> <p>Note: If you are using the <i>*/xx</i>' format, the smallest value for 'xx' is 15.</p> <p>You can enter <i>*/(15-59)</i> or a range such as 23-43. An asterisk (*) is not allowed.</p>
hour	<p>This parameter may contain either an asterisk, (*), which implies "run every hour," or a number value between 0-23.</p> <p>You can enter <i>*/(0-23)</i>, a range such as 12-21, or just the *.</p>

day_of_the_month	<p>This parameter may contain either an asterisk, (*), which implies "run every day of the month," or a number value between 1-31.</p> <p>You can enter */(1-31), a range such as 3-22, or just the *.</p>
month	<p>This parameter may contain either an asterisk, (*), which implies "run every month," or a number value between 1-12.</p> <p>You can enter */(1-12), a range such as 1-5, or just the *. You can also enter the first three letters of any month (must use lowercase letters).</p>
day_of_the_week	<p>This parameter may contain either an asterisk (*), which implies "run every day of the week," or a numeric value between 0-6. Crontab interprets 0 as Sunday. You can also enter the first three letters of the week (must use lowercase letters).</p>

For example, the following command creates a schedule `schedule1` for automated snapshot creation of the `fs1` file system every 3 hours each day, and maintains only 30 snapshots:

```
Storage> snapshot schedule create schedule1 fs1 30 0 */3 * * *
```

To modify a snapshot schedule

- ◆ To modify a snapshot schedule, enter the following:

```
Storage> snapshot schedule modify schedule_name
      fs_name
      max_snapshot_limit
      minute [hour]
      [day_of_the_month] [month] [day_of_the_week]
```

For example, to modify the existing schedule so that a snapshot is created at 2:00 am on the first day of the week, enter the following:

```
Storage> snapshot schedule modify schedule1 fs1 *2**1
```

To display a snapshot schedule

- ◆ To display all of the schedules for automated snapshots, enter the following:

```
Storage> snapshot schedule show [fs_name] [schedule_name]
```

<code>fs_name</code>	Displays all of the schedules of the specified file system. If no file system is specified, schedules of all of the file systems are displayed.
----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------

<code>schedule_name</code>	Displays the schedule name. If no schedule name is specified, then all of the schedules created under <i>fs_name</i> are displayed.
----------------------------	-------------------------------------------------------------------------------------------------------------------------------------

Managing automated snapshots

You can remove all of the automated snapshots created by a schedule, specify that certain snapshots be preserved, or delete a schedule for a file system.

To remove all snapshots

- ◆ To automatically remove all of the snapshots created under a given schedule and file system name (excluding the preserved and online snapshots), enter the following:

```
Storage> snapshot schedule destroyall schedule_name
fs_name
```

The `destroyall` command only destroys snapshots that are offline. If some of the snapshots in the schedule are online, the command exists at the first online snapshot.

Note: The `Storage> snapshot schedule destroyall` command may take a long time to complete depending on how many snapshots are present that were created using schedules.

Preserved snapshots are never destroyed automatically or as part of the `destroyall` command.

Example 1: If you try to destroy all automated snapshots when two of the automated snapshots are still mounted, Veritas Access returns an error. No snapshots under the given schedule and file system are destroyed.

```
Storage> snapshot schedule destroyall schedule1 fs1
ACCESS snapshot ERROR V-288-1074 Cannot destroy snapshot(s)
schedule1_7_Dec_2009_17_58_02_UTC schedule1_7_Dec_2009_16_58_02_UTC
in online state.
```

Example 2: If you try to destroy all automated snapshots (which are in an offline state), the operation completes successfully.

```
Storage> snapshot schedule destroyall schedule2 fs1
100% [#] Destroy automated snapshots
```

To preserve snapshots

- ◆ To preserve the specified snapshots corresponding to an existing schedule and specific file system name, enter the following:

```
Storage> snapshot schedule preserve schedule_name
fs_name snapshot_name
```

snapshot_name is a comma-separated list of snapshots..

To delete a snapshot schedule

- ◆ To delete a snapshot schedule, enter the following:

```
Storage> snapshot schedule delete fs_name [schedule_name]
```

Using instant rollbacks

This chapter includes the following topics:

- [About instant rollbacks](#)
- [Creating a space-optimized rollback](#)
- [Creating a full-sized rollback](#)
- [Listing Veritas Access instant rollbacks](#)
- [Restoring a file system from an instant rollback](#)
- [Refreshing an instant rollback from a file system](#)
- [Bringing an instant rollback online](#)
- [Taking an instant rollback offline](#)
- [Destroying an instant rollback](#)
- [Creating a shared cache object for Veritas Access instant rollbacks](#)
- [Listing cache objects](#)
- [Destroying a cache object of a Veritas Access instant rollback](#)

About instant rollbacks

Instant rollbacks are volume-level snapshots. All rollback commands take a file system name as an argument and perform operations on the underlying volume of that file system.

Note: If you plan to add a tier to the file system, add the tier first and then create the rollback. If you add the tier after a rollback exists, the rollback hierarchy would have inconsistencies because the rollback is not aware of the tier.

Both space-optimized and full-sized rollbacks are supported by Veritas Access. Space-optimized rollbacks use a storage cache, and do not need a complete copy of the original volume's storage space. However, space-optimized rollbacks are not suitable for write-intensive volumes, because the copy-on-write mechanism may degrade the performance of the volume. Full-sized rollbacks use more storage, but that has little impact on write performance after synchronization is completed.

Both space-optimized rollbacks and full-sized rollbacks can be used instantly after operations such as create, restore, or refresh.

Note: When instant rollbacks exist for a volume, you cannot disable the FastResync option for a file system.

When creating instant rollbacks for volumes bigger than 1T, there may be error messages such as the following:

```
ACCESS instant_snapshot ERROR V-288-1487 Volume prepare for full-fs1-1
failed.
```

An error message may occur because the default amount of memory allocated for a Data Change Object (DCO) may not be large enough for such big volumes. You can use the `vxtune` command to change the value. The default value is 6M, which is the memory required for a 1T volume.

To change it to 15M, use the following command:

```
vxtune volpagemod_max_memsz `expr 15 \* 1024 \* 1024`
```

Creating a space-optimized rollback

To create a space-optimized rollback

- ◆ To create a space-optimized rollback for a specified file system, enter the following:

```
Storage> rollback create space-optimized rollback_name  
fs_name [cacheobj]
```

<code>rollback_name</code>	Indicates the name of the rollback.
<code>fs_name</code>	Indicates the name of the file system for which to create the space-optimized rollback.
<code>cacheobj</code>	Indicates the cache object name. If the cache object is specified, then the shared cache object is used. Otherwise, Veritas Access automatically creates a cache object for the rollback.

Creating a full-sized rollback

To create a full-sized rollback for a specified file system

- ◆ To create a Veritas Access full-sized rollback for a specified file system, enter the following:

```
Storage> rollback create full-sized rollback_name  
fs_name pool
```

<code>rollback_name</code>	Indicates the name of the rollback.
<code>fs_name</code>	Indicates the name of the file system for which to create the full-sized rollback.
<code>pool</code>	Indicates the name of the pool on which to create the full-sized rollback. The disks used for the rollback are allocated from the specified pool.

Listing Veritas Access instant rollbacks

To list Veritas Access instant rollbacks

- ◆ To list Veritas Access instant rollbacks, enter the following:

```
Storage> rollback list [fs_name]
```

where *fs_name* is the name of the file system where you want to list the instant rollbacks.

If no file system is specified, instant rollbacks are displayed for all the file systems.

Restoring a file system from an instant rollback

Prior to restoring a file system by a specified rollback, the file system should be offline.

See [“Taking an instant rollback offline”](#) on page 452.

To restore a file system from an instant rollback

- 1 To restore a file system from an instant rollback, enter the following:

```
Storage> rollback restore fs_name rollback_name
```

fs_name Indicates the name of the file system that you want to restore.

rollback_name Indicates the name of the rollback that you want to restore.

- 2 Bring the file system back online.

See [“Bringing an instant rollback online”](#) on page 452.

Bringing the file system online may take some time depending on the size of the file system.

Refreshing an instant rollback from a file system

To refresh an instant rollback from a file system

- ◆ To refresh an instant rollback from a file system, enter the following:

```
Storage> rollback refresh rollback_name fs_name
```

rollback_name Indicates the name of the rollback that you want to refresh.

fs_name Indicates the name of the file system that you want to refresh.

Bringing an instant rollback online

You can choose to bring an instant rollback online and use it as a live file system. If the original file system is offline for some reason, the instant rollback can be used as a backup.

When an instant rollback is mounted and written to with new data, the instant rollback may no longer be suitable for use in restoring the contents of the original volume. If you chose to write to an instant rollback, create another instant rollback as a backup of the original file system.

Bringing an instant rollback online

- ◆ To bring an instant rollback online, enter the following:

```
Storage> rollback online rollback_name
```

rollback_name Indicates the name of the rollback that you want to bring online.

The instant rollback is available for read/write access just as the file system.

Taking an instant rollback offline

Taking an instant rollback offline

- ◆ To take an instant rollback offline, enter the following:

```
Storage> rollback offline rollback_name
```

rollback_name Indicates the name of the rollback that you want to take offline.

For example:

Destroying an instant rollback

The instant rollback must be in the offline state before it can be destroyed.

See [“Taking an instant rollback offline”](#) on page 452.

To destroy an instant rollback

- ◆ To destroy an instant rollback, enter the following:

```
Storage> rollback destroy rollback_name fs_name
```

rollback_name Indicates the name of the rollback that you want to destroy.

fs_name Indicates the name of the file system that you want to destroy.

For example:

Creating a shared cache object for Veritas Access instant rollbacks

You can create a shared cache object for Veritas Access instant rollbacks.

Space-optimized rollbacks use a storage cache to save the data. Using a shared cache object, cache storage can be shared by all the space-optimized rollbacks.

To create a shared cache object for Veritas Access instant rollbacks

- ◆ To create a shared cache object for Veritas Access instant rollbacks, enter the following:

```
Storage> rollback cache create cache_name [cache_size] [pool]
```

cache_name	Indicates the name of the cache object you want to create.
cache_size	<p>Indicates the cache size for the cache object. Cache size can be specified in any units, such as M, G, or T.</p> <p>The size of the shared cache object should be sufficient to record changes to the file system during intervals between instant rollback refreshes. By default, the size of the cache object for an instant rollback is 20% of the total size of the parent file system.</p> <p>The size of the cache object is dependent on your environment.</p>
pool	<p>Indicates the pool for storing the cache object.</p> <p>For better performance, the pool used for the space-optimized rollback should be different from the pool used by the file system.</p>

To convert an existing file system into a cache object

- 1 Select or create a file system with the layout that you want to use for the cache object. In this way, you can create cache objects with any kind of file system type. If you use an existing file system, the data on the file system is lost when you convert it to a cache object.

The following example shows how to create a file system with a file system type of striped:

```
Storage> fs create striped cobj1 100m 2 pool0
100% [#] Creating striped filesystem
```

- 2 Run the `Storage> rollback cache create` command without the `cache_size` and `pool` parameters:

```
Storage> rollback cache create cache_name
```

`cache_name` is the name of the file system from step 1 that you want to convert.

A confirmation message in the Veritas Access CLI asks if you want to convert the specified file system to a cache object.

For example, to convert the striped file system `cobj1` to a cache object:

```
Storage> rollback cache create cobj1
ACCESS rollback WARNING V-288-0 Filesystem cobj1 will be converted to
cache object.
All data on Filesystem cobj1 will be lost
ACCESS rollback WARNING V-288-0 Are you sure you want to convert cobj1
to a cache object? (yes/no)
yes
100% [#]
```

- 3 Verify that the new cache object exists:

```
Storage> rollback cache list
```

CACHE NAME	TOTAL (Mb)	USED (Mb) (%)	AVAIL (Mb) (%)	SDCNT
cache1	15	15 (100)	0 (0)	2
cobj1	100	4 (4)	96 (96)	0

Listing cache objects

The `Storage> rollback cache list` command allows you to list the Veritas Access instant rollbacks that are using a cache object.

To list cache objects for Veritas Access instant rollbacks

- ◆ To list cache objects for Veritas Access instant rollbacks, enter the following:

```
Storage> rollback cache list [cache_name]
```

where *cache_name* is the name of the cache object you want to display. When *cache_name* is specified, the instant rollbacks that are using the cache object are listed.

A disabled cache object is listed with '-' as the attribute. *cache2* and *mycache* are in the DISABLED state.

For example:

```
Storage> rollback cache list
```

CACHE NAME	TOTAL (Mb)	USED (Mb)	(%)	AVAIL (Mb)	(%)	SDCNT
cache1	15	15	(100)	0	(0)	2
cobj1	100	4	(4)	96	(96)	0
cache2	-	-	-	-	-	-
mycache	-	-	-	-	-	-

SDCNT is the number of subdisks that have been created on the cache object.

If the cache object is disabled for some reason, it will automatically be restarted when the `Storage> rollback cache list cache_name` command is run.

For example:

```
Storage> rollback cache list cache2
rollbacks located on cache cache2:
roll3
ACCESS rollback WARNING V-288-0 Cache object cache2 was DISABLED,
trying to restart it.
ACCESS rollback INFO V-288-0 Cache object cache2 started successfully.
```

You can choose to start the cache object, or destroy it after destroying all the instant rollbacks located on it.

See [“Destroying a cache object of a Veritas Access instant rollback”](#) on page 457.

If you did not assign a cache object, a cache object is internally created for the instant rollback.

Destroying a cache object of a Veritas Access instant rollback

To destroy a cache object of a Veritas Access instant rollback

- ◆ To destroy a cache object of a Veritas Access instant rollback, enter the following:

```
Storage> rollback cache destroy cache_name
```

where *cache_name* is the name of the cache object that you want to destroy.

You can only destroy the cache object if there is no instant rollback that is using this cache object.

Configuring Veritas Access with the NetBackup client

This chapter includes the following topics:

- [About Veritas Access as a NetBackup client](#)
- [Prerequisites for configuring the NetBackup client](#)
- [About the NetBackup Snapshot Client](#)
- [About NetBackup snapshot methods](#)
- [Enabling or disabling the NetBackup SAN client](#)
- [Workflow for configuring Veritas Access for NetBackup](#)
- [Registering a NetBackup master server or adding an optional media server](#)
- [Displaying the excluded files from backup](#)
- [Displaying the included and excluded files for backups](#)
- [Adding or deleting patterns to the list of files in backups](#)
- [Configuring or resetting the virtual IP address used by NetBackup](#)
- [Configuring the virtual name of NetBackup](#)
- [Displaying the status of NetBackup services](#)
- [Configuring backup operations using NetBackup or other third-party backup applications](#)

- Performing a backup or restore of a Veritas Access file system over a NetBackup SAN client
- Performing a backup or restore of a snapshot
- Installing or uninstalling the NetBackup client
- Configuring Veritas Access for NetBackup cloud storage

About Veritas Access as a NetBackup client

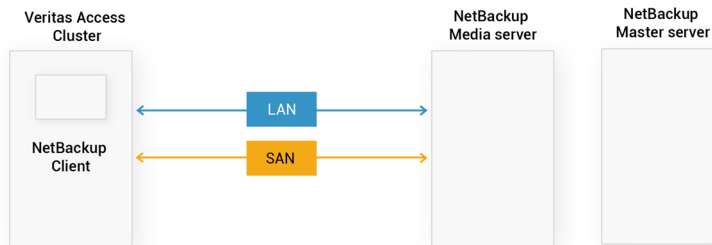
Veritas Access is integrated with Veritas NetBackup so that a NetBackup administrator can back up your Veritas Access file systems to NetBackup master or media servers and retain the data as per your company policy. Once data is backed up, a storage administrator can delete unwanted data from Veritas Access. The NetBackup master and media servers that run on separate computers from Veritas Access are licensed separately from Veritas Access.

You configure NetBackup domain information using any one of the following Veritas Access interfaces:

- CLISH
The Veritas Access CLISH has a dedicated `Backup>` menu. From the `Backup>` menu, register the NetBackup client with the NetBackup domain. Information is saved in the `bp.conf` file on Veritas Access.
- GUI
Settings > NetBackup Configuration
See the online Help for how to configure NetBackup using the GUI.
- RESTful APIs
See the *Veritas Access RESTful API Guide*.

Consolidating storage reduces the administrative overhead of backing up and restoring many separate file systems. Critical file data can be backed up and restored through the NetBackup client on Veritas Access.

Figure 30-1 Configuration of Veritas Access with NetBackup



Prerequisites for configuring the NetBackup client

Before configuring the NetBackup client for Veritas Access, you must have completed the following:

- You must have a NetBackup master server external to your Veritas Access cluster. The NetBackup administrator configures the NetBackup master server. See the *NetBackup* product documentation for more information.
- Add the valid licenses on the NetBackup master server.
- Make sure that host names of the master server and the NetBackup client installed on Veritas Access can be resolved and the reverse lookup is working.

About the NetBackup Snapshot Client

A snapshot is a point-in-time, read-only, disk-based copy of a client volume. After the snapshot is created, NetBackup backs up data from the snapshot, not directly from the client's primary or original volume. Users and client operations can access the primary data without interruption while data on the snapshot volume is backed up. The contents of the snapshot volume are cataloged as if the backup was produced directly from the primary volume. After the backup is complete, the snapshot-based backup image on storage media is indistinguishable from a traditional, non-snapshot backup.

About NetBackup snapshot methods

NetBackup can create different types of snapshots. Each snapshot type that you configure in NetBackup is called a snapshot method. Snapshot methods enable

NetBackup to create snapshots within the storage stack (such as the file system) where the data resides. When the data resides in a file system, NetBackup can use a file system method, depending on the client operating system and the file system type.

You select the snapshot method in the backup policy as explained in the *Veritas NetBackup Snapshot Client Administrator's Guide*.

Note: When using Veritas Access with NetBackup, select the `VxFs_Checkpoint` snapshot method. The checkpoint feature is not supported on a scale-out file system.

Enabling or disabling the NetBackup SAN client

You can enable or disable the NetBackup SAN client on Veritas Access. The NetBackup SAN client should only be enabled on Veritas Access if the required licenses are installed on the NetBackup Master Server. If you do not have the required license for the NetBackup SAN client, then you must disable the SAN client on Veritas Access. Otherwise, the Veritas Access backup service fails to start.

To enable or disable the NetBackup SAN client

- ◆ To enable or disable the NetBackup SAN client, enter the following:

```
Backup> netbackup sanclient enable | disable
```

enable	Enables the NetBackup SAN client.
--------	-----------------------------------

disable	Disables the NetBackup SAN client.
---------	------------------------------------

```
Backup> netbackup sanclient enable
Success.
```

Workflow for configuring Veritas Access for NetBackup

To back up your data with NetBackup, you must register the installed and configured NetBackup master server with Veritas Access.

To configure NetBackup for Veritas Access, perform the following tasks in the order shown:

Make sure that the prerequisites are met.	See “Prerequisites for configuring the NetBackup client” on page 460.
Register the NetBackup master server.	See “Registering a NetBackup master server or adding an optional media server” on page 463.
Display the current status of the NetBackup client.	See “Displaying the status of NetBackup services” on page 467.
Reset the values for the NetBackup master server.	See “Registering a NetBackup master server or adding an optional media server” on page 463.
Display the current status of the NetBackup client.	See “Displaying the status of NetBackup services” on page 467.
Reset the NetBackup virtual name.	See “Configuring the virtual name of NetBackup” on page 466.
Register the NetBackup master server with the NetBackup client.	See “Registering a NetBackup master server or adding an optional media server” on page 463.
Configure the virtual name that the NetBackup master server uses for the NetBackup client.	See “Configuring the virtual name of NetBackup” on page 466.
Display the current status of the NetBackup client.	See “Displaying the status of NetBackup services” on page 467.
Verify that Veritas Access is configured with the NetBackup client.	See “Displaying the status of NetBackup services” on page 467.
Specify the files to back up or restore.	See “Performing a backup or restore of a Veritas Access file system over a NetBackup SAN client” on page 469.
Specify the snapshot to back up or restore	See “Performing a backup or restore of a snapshot” on page 470.
Uninstalling or installing the NetBackup client.	See “Installing or uninstalling the NetBackup client” on page 470.
Configuring Veritas Access for NetBackup cloud storage	See “Configuring Veritas Access for NetBackup cloud storage” on page 474.

Registering a NetBackup master server or adding an optional media server

You register the NetBackup master server so that it can communicate with Veritas Access. If necessary, you can reset the values of the NetBackup master server to its default configurations. You can optionally add a media server.

The NetBackup master server can be the NetBackup media server, but it is not mandatory that the NetBackup master server be the NetBackup media server. In production environments, the NetBackup media server is separate from the NetBackup master server.

See the `backup_netbackup(1)` man page for detailed examples.

To register the NetBackup master server with Veritas Access

- ◆ Register the NetBackup master server with Veritas Access.

```
Backup> netbackup master-server set server
```

To reset the value for the NetBackup master server

- ◆ Reset the value for the NetBackup master server.

```
Backup> netbackup master-server reset
```

To add an optional NetBackup media server

- ◆ Add an optional NetBackup media server.

If the NetBackup master server is also acting as a NetBackup media server, then add the NetBackup media server using the NetBackup master server hostname.

For example:

```
Backup> netbackup media-server add FQDN of master server
```

To delete an already configured NetBackup media server

- ◆ Delete an already configured NetBackup media server.

```
Backup> netbackup media-server delete server
```

Displaying the excluded files from backup

To display the entries in the excluded list from backup

- ◆ Display the entries in the excluded list from backup.

```
Backup> netbackup exclude_list show [policy] [schedule]
```

policy Lists the excluded entries by specifying a NetBackup policy.

schedule If a NetBackup policy schedule is specified, then the excluded list entries for the specified NetBackup policy and NetBackup policy schedule are displayed.

```
Backup> netbackup exclude_list show
```

Pattern	Policy	Schedule
-----	-----	-----
hosts	-	-
iscsid.conf	-	-
iscsid.conf	NBU_access12	-
/vx/fs100/as*	policy	-
/vx/fs100/*mp3	policy	-
/vx/fs200/bs*	policy2	sched

The hyphens in the command output indicate that no values have been entered.

Displaying the included and excluded files for backups

You can specify a policy pattern that lets you specify which files to include or exclude from NetBackup backups. For example, you can specify that only `.gif` files are backed up, and `.iso` files are excluded. You can then display those files.

See the `backup_netbackup(1)` man page for detailed examples.

To display files included or excluded for backups

- ◆ Display the files that are included or excluded for backups.

For included files:

```
Backup> netbackup include_list show [policy] [schedule]
```

For excluded files:

```
Backup> netbackup exclude_list show [policy] [schedule]
```

Adding or deleting patterns to the list of files in backups

You can add or delete specified patterns to or from the files that you want to include or exclude from NetBackup backups. For example, you can create a backup policy with different patterns such that only `.gif` files are backed up and `.iso` files are excluded.

See the `backup_netbackup(1)` man page for detailed examples.

To add or delete the given pattern to the list of files included for backup

- ◆ Add the specified pattern to the files that are included for backup.

For adding specified patterns to included files:

```
Backup> netbackup include_list add pattern [policy] [schedule]
```

For deleting specified patterns from included files:

```
Backup> netbackup include_list delete pattern [policy] [schedule]
```

To add or delete a given pattern to the list of files excluded from backup

- ◆ Add a given pattern to the list of files that are excluded from backup.

For adding a given pattern to excluded files:

```
Backup> netbackup exclude_list add pattern [policy] [schedule]
```

For deleting the given pattern from excluded files:

```
Backup> netbackup exclude_list delete pattern [policy] [schedule]
```

Configuring or resetting the virtual IP address used by NetBackup

You can configure or reset the virtual IP address of NetBackup. This address is a highly-available virtual IP address in the cluster.

Note: Configure the virtual IP address using the `Backup> virtual-ip set` command so that it is different from all of the virtual IP addresses, including the console server IP address and the physical IP addresses that are used to install Veritas Access. Use the `Network> ip addr show` command to display the currently assigned virtual IP addresses on Veritas Access.

See the `backup_virtual-ip(1)` man page for detailed examples.

To configure or reset the virtual IP address used by NetBackup

- 1 Configure the virtual IP address of NetBackup on Veritas Access.

```
Backup> virtual-ip set ipaddr [device]
```

- 2 Reset the virtual IP address of NetBackup on Veritas Access.

```
Backup> virtual-ip reset
```

See [“Configuring the virtual name of NetBackup”](#) on page 466.

Configuring the virtual name of NetBackup

You can either configure the virtual name for the NetBackup master server, or you can reset the value to its default or unconfigured state.

See the `backup_virtual-name(1)` man page for detailed examples.

To set or reset the NetBackup virtual name

- ◆ Set or reset the NetBackup virtual name.

For setting the virtual name:

```
Backup> virtual-name set name
```

For resetting the virtual name:

```
Backup> virtual-name reset
```

Make sure that *name* can be resolved through DNS, and its IP address can be resolved back to *name* through the DNS reverse lookup. Also, make sure that *name* resolves to an IP address that is configured by using the `Backup> virtual-ip` command.

See [“Configuring or resetting the virtual IP address used by NetBackup”](#) on page 466.

Displaying the status of NetBackup services

Use the backup commands to display the status of the NetBackup services.

See the following man pages for detailed examples:

- `backup_show(1)`
- `backup_status(1)`
- `backup_start(1)`
- `backup_stop(1)`

To display NetBackup services

- ◆ Display the current NetBackup services.

```
Backup> show
```

If the settings were configured while the backup and the restore services were online, they may not be in use by Veritas Access. To display all of the configured settings, first run the `Backup> stop` command, then run the `Backup> start` command.

To display the status of backup services

- ◆ Display the status of backup services.

```
Backup> status
```

If the NetBackup server is started and online, then `Backup> status` displays any on-going backup or restore jobs.

To start or stop backup services

- 1 Start the backup services.

```
Backup> start [nodename]
```

You can also change the status of a virtual IP address to online after it has been configured using the `Backup> virtual-ip` command. This command applies to any currently active node in the cluster that handles backup and restore jobs.

See [“Configuring or resetting the virtual IP address used by NetBackup”](#) on page 466.

- 2 Stop the backup services.

```
Backup> stop
```

You can also change the status of a virtual IP address to offline after it has been configured using the `Backup> virtual-ip` command.

See [“Configuring or resetting the virtual IP address used by NetBackup”](#) on page 466.

Configuring backup operations using NetBackup or other third-party backup applications

You can backup Veritas Access using NetBackup client capability, or backup applications from other third-party companies that use the standard NFS mount to backup over the network.

For information on NetBackup, refer to the *NetBackup* product documentation set.

The Backup commands configure the local NetBackup installation of Veritas Access to use an external NetBackup master server or media server. When NetBackup is installed on Veritas Access, it acts as a NetBackup client to perform IP-based backups of Veritas Access file systems.

Note: A new public IP address, not an IP address that is currently used, is required for configuring the NetBackup client. Use the `Backup> virtual-ip` and `Backup> virtual-name` commands to configure the NetBackup client.

Performing a backup or restore of a Veritas Access file system over a NetBackup SAN client

You can perform a backup or restore of a Veritas Access file system over a NetBackup SAN client. A NetBackup SAN client is a NetBackup client for which Fibre Transport services are activated.

Backup and restore operations are done on the NetBackup master server by a NetBackup administrator using the NetBackup Administration Console. If the NetBackup master server can connect to the NetBackup client on Veritas Access, the NetBackup master server starts the backup or restore operations.

Before performing a backup or restoration of a Veritas Access file system over a NetBackup SAN client, do the following:

- Verify that the virtual IP address is online.
- Verify that the NetBackup client state is online.
- Configure the Fibre Transport media server.
 See the *Veritas NetBackup SAN client and Fibre Transport Guide* for more information on configuring the NetBackup Fibre Transport media server.

See the `backup(1)` man page for detailed examples.

To perform a backup of a file system over a NetBackup SAN client

- 1 Check the status of the NetBackup client.

```
Backup> status
```

- 2 Enable the SAN client from the CLI.

```
Backup> netbackup sanclient enable
```

- 3 Verify if the SAN client has been enabled or not from the CLI.

```
Backup> status
```

- 4 Using the NetBackup Administration Console, start a backup operation.

To perform a restore of a file system over a NetBackup SAN client

- 1 Check the status of the NetBackup client.

```
Backup> status
```

- 2 Using the NetBackup Administration Console, start a restore operation.
- 3 Check the status of the NetBackup client.

```
Backup> status
```

Performing a backup or restore of a snapshot

Using the NetBackup Administration Console, a NetBackup administrator can perform a backup or restore of a snapshot.

Veritas Access as a NetBackup client supports the VxFS_Checkpoint snapshot method. See the *Veritas NetBackup Snapshot Client Administrator's Guide* for more information on configuring snapshot policies.

To perform a backup of a snapshot

- ◆ Using the NetBackup Administration Console, start a snapshot backup.
The snapshot triggered by the NetBackup job can be seen from the CLI.

```
Storage> snapshot list
```

To perform a restore of a snapshot

- 1 Using the NetBackup Administration Console, navigate to **Backup, Archive, and Restore**.
- 2 Click the **Restore Files** tab.
- 3 Click the **Restore** option.
- 4 Specify the directory location for performing the restore operation.

```
/vx/name_of_file_system/Restores
```

Installing or uninstalling the NetBackup client

The NetBackup master server version should be higher or equal to the NetBackup client version. To upgrade the NetBackup client, uninstall the currently installed version of the NetBackup client and then install the specified version of the

NetBackup client. The uninstall command runs on all the nodes of the Veritas Access cluster.

Veritas Access supports two major versions of the NetBackup client, version 7.7 and 8.0. By default, Veritas Access comes with the 8.0 version of the NetBackup client.

See the `backup (1)` man page for detailed examples.

To install the NetBackup client

- 1 Display the currently installed version of the NetBackup client.

```
Backup> show
```

- 2 Install the specified NetBackup client.

```
Backup> install version [URL]
```

You must specify the version of the NetBackup client that you want to install. If you do not specify a URL, the `Backup> install` command has the information on the file system for the location it needs. Specify the major release version (8.0 or 7.7) as the second parameter. You can specify the NetBackup package minor release or patches (7.7.1 for a 7.7 major release) as the third parameter to install.

If the base NetBackup client version is 7.7.

```
Backup> install 7.7 scp://support@192.168.2.10:/home/NetBackup_7.7_
CLIENTS2.tar.gz
```

If the base NetBackup client version is 8.0.

```
Backup> install 8.0 scp://support@192.168.2.10:/home/NetBackup_8.0_
CLIENTS2.tar.gz
```

If there are minor releases or patches from the NetBackup client.

```
Backup> install 7.7 scp://support@192.168.2.10:/home/NetBackup_7.7.1_
CLIENTS2.tar.gz
```

```
Backup> install 7.7 scp://support@192.168.2.10:/home/NetBackup_7.7.2_
CLIENTS2.tar.gz
```

For example, consider that the NetBackup client binaries are placed on the following host:

```
192.168.2.10
```

```
Backup> install 7.7 scp://support@192.168.2.10:/home/NetBackup_7.7.3_
CLIENTS2.tar.gz
```

Where 192.168.2.10 is the host IP address on which the NetBackup client packages are placed.

```
NetBackup_7.7.3_CLIENTS2.tar.gz
```

This is the NetBackup client package.

- 3** Double check if the Red Hat compatible NetBackup client is available in this package.

```
support:  
system user and specify password when prompted.
```

- 4** Verify that the specified NetBackup client is installed.

```
Backup> show
```

To uninstall the NetBackup client

- 1** Display the currently installed version of the NetBackup client.

```
Backup> show
```

- 2** Uninstall the existing version of the NetBackup client.

```
Backup> uninstall
```

- 3** Display the current running version of the NetBackup client.

```
Backup> show
```

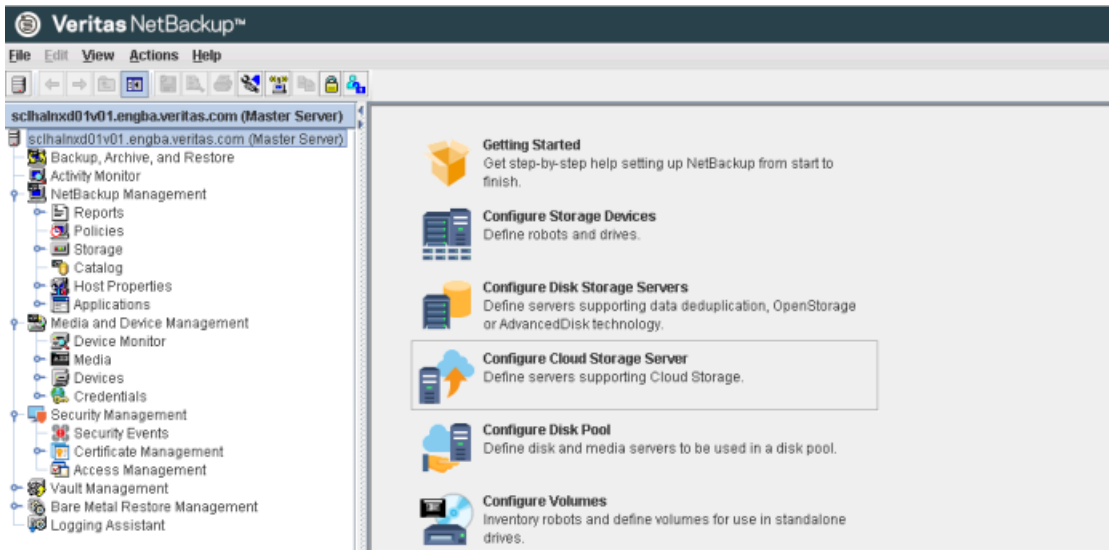
- 4** Verify that the NetBackup client is not installed.

```
Backup> show
```

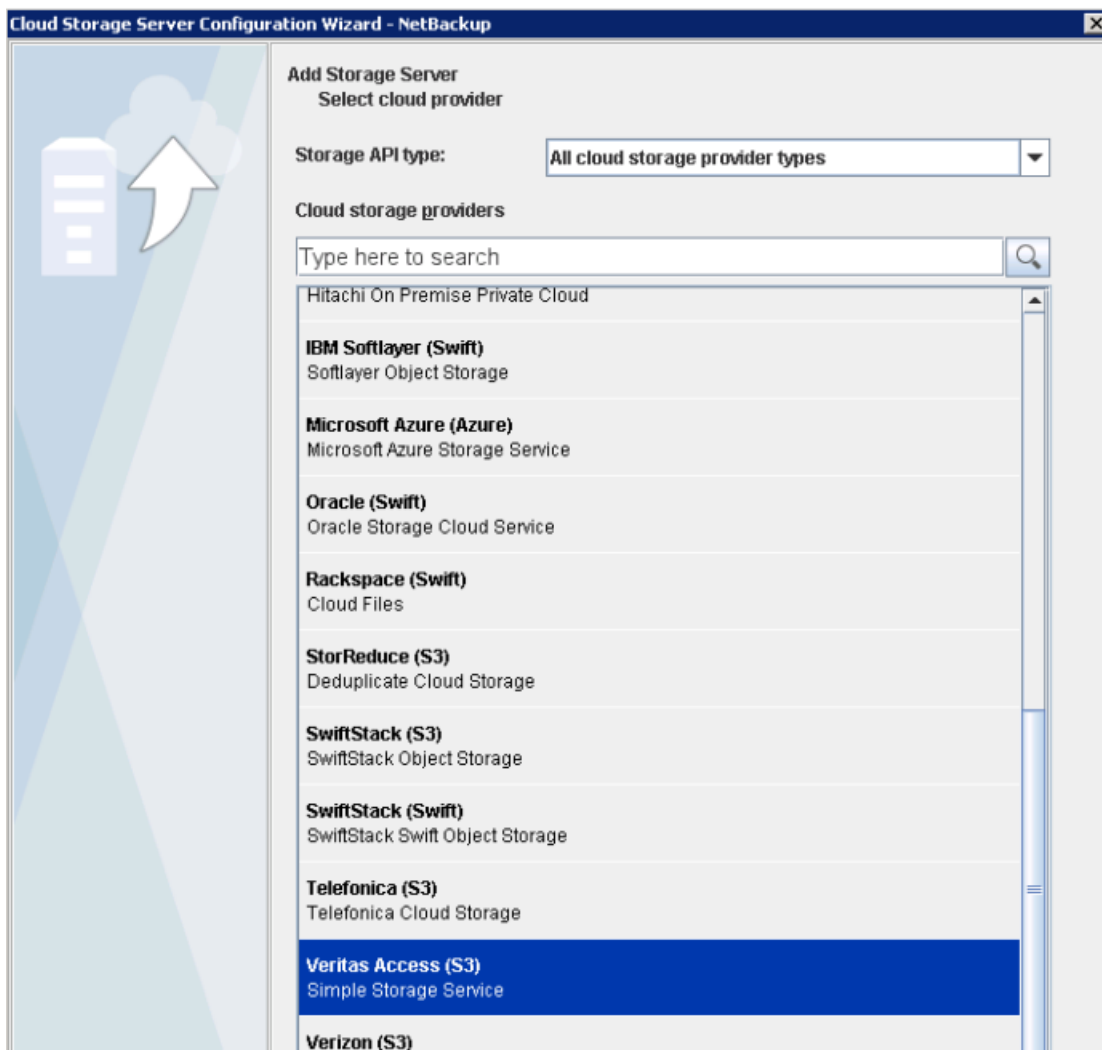
Configuring Veritas Access for NetBackup cloud storage

To configure Veritas Access for NetBackup cloud storage

- 1 Log on to the NetBackup Console and select **Configure Cloud Storage Server**.



- 2 From the **Cloud Storage Server Configuration** wizard, select **Veritas Access (S3)**.



- 3 Add the required information such as **Service host**, **Service endpoint**, **HTTP/HTTPS port**, **Access key ID**, **Secret access key**, and follow the rest of the wizard prompts.

Cloud Storage Server Configuration Wizard - NetBackup

Add Storage Server
 Select a media server and provide cloud storage service credentials. To be listed below in the media server drop-down list a security certificate must be deployed and NetBackup must be running including the NetBackup CloudStore Service Container (nbcssc).

Cloud storage provider: Veritas Access

Service host: s3.nbappclust

Storage server name:

Add Cloud Storage

Media server name: schahed01v01.nbgba.veritas.com

Deduplication
☐ Deduplicate data using MSDP before writing to cloud storage

Local cache directory:

Access details for Veritas Access account

Access key ID:

Secret access key:

Add Cloud Storage

General Settings | Region Settings

Provider type: Veritas Access

Service host:

Service endpoint:

HTTP port: 80

HTTPS port: 443

Storage server name:

Endpoint access style: Path Style

OK **Cancel** **Help**

If you do not have Veritas Access account
[Create an account with Veritas Access.](#)

Reference

- [Appendix A. Veritas Access documentation](#)
- [Appendix B. Veritas Access tuning](#)

Veritas Access documentation

This appendix includes the following topics:

- [Using the Veritas Access product documentation](#)
- [About accessing the online man pages](#)

Using the Veritas Access product documentation

The latest version of the Veritas Access product documentation is available on the Veritas Services and Operations Readiness Tools (SORT) website.

<https://sort.veritas.com/documents>

You need to specify the product and the platform and apply other filters for finding the appropriate document.

Make sure that you are using the current version of documentation. The document version appears on page 2 of each guide. The publication date appears on the title page of each document. The documents are updated periodically for errors or corrections.

The following guides are available for the Access Appliance:

- *Veritas Access Appliance Initial Configuration and Administration Guide*

Note: This guide replaces the *Veritas Access Appliance Getting Started Guide*

- *Veritas Access Appliance Commands Reference Guide*
- *Veritas Access 3340 Appliance Product Description*

- *Veritas Access 3340 Appliance Hardware Installation Guide*
- *Veritas Appliance AutoSupport 2.0 Reference Guide*
- *Veritas Access Safety and Maintenance Guide*

The following Veritas Access 7.3.1 documents are available on the SORT site:

- *Veritas Access Administrator's Guide*
- *Veritas Access Command Reference Guide*
- *Veritas Access NetBackup Solutions Guide*
- *Veritas Access Release Notes*
- *Veritas Access RESTful API Guide*
- *Veritas Access Third-Party License Agreements*
- *Veritas Access Troubleshooting Guide*
- *Veritas Access Enterprise Vault Solutions Guide*

About accessing the online man pages

You access the online man pages by typing `man name_of_command` at the command line.

The example shows the result of entering the `Network> man ldap` command.

```
Network> man ldap
```

```
NAME
```

```
ldap - configure LDAP client for authentication
```

```
SYNOPSIS
```

```
ldap enable
```

```
ldap disable
```

```
ldap show [users|groups|netgroups]
```

```
ldap set {server|port|basedn|binddn|ssl|rootbinddn|users-basedn|  
groups-basedn|netgroups-basedn|password-hash} value
```

```
ldap get {server|port|basedn|binddn|ssl|rootbinddn|  
users-basedn|groups-basedn|netgroups-basedn|password-hash}
```

You can also type a question mark (?) at the prompt for a list of all the commands that are available for the command mode that you are in. For example, if you are within the `admin` mode, if you type a question mark (?), you will see a list of the available commands for the `admin` mode.

```
ACCESS> admin ?
```

```
Entering admin mode...
```

```
ACCESS.Admin>
```

```
exit          --return to the previous menus
logout        --logout of the current CLI session
man           --display on-line reference manuals
passwd        --change the administrator password
show          --show the administrator details
supportuser   --enable or disable the support user
user          --add or delete an administrator
```

To exit the command mode, enter the following: `exit`.

For example:

```
ACCESS.Admin> exit
```

```
ACCESS>
```

To exit the system console, enter the following: `logout`.

For example:

```
ACCESS> logout
```


Veritas Access tuning

This appendix includes the following topics:

- [File system mount-time memory usage](#)

File system mount-time memory usage

Mounting a file system on a computer system allocates system memory that is not freed until the file system is unmounted. The amount of memory allocated at mount time is directly proportional to the size of the file system being mounted. The amount of memory that is allocated at mount-time is therefore important information to help determine the system memory requirements for a Veritas Access environment. The mount-time memory requirement is different if you expect to mount a total of 1 PB of storage or 2 PBs of storage. The number of files currently in the file system does not affect the amount of memory allocated at mount-time. The amount of memory allocated at mount-time is also inversely proportional to the file system block size.

The information required to determine the amount of memory allocated at mount time is the total size of all the file systems that are mounted on the same computer system at the same time and the block size of each file system.

The amount of memory allocated at mount time can therefore be estimated by obtaining the total size of all the file systems that are mounted on a system according to the file system block size. So four totals in all, one for each file system block size of 1 KB, 2 KB, 4 KB, and 8 KB.

Table B-1 File system mount-time memory usage

File system block size	Total size of mounted file systems	Memory allocation at mount time
1 KB	'a' TBs	'w'MBs allocated per TB
2 KB	'b' TBs	'x'MBs allocated per TB

Table B-1 File system mount-time memory usage (*continued*)

File system block size	Total size of mounted file systems	Memory allocation at mount time
4 KB	'c' TBs	'y' MBs allocated per TB
8 KB	'd' TBs	'z' MBs allocated per TB

The mount-time memory requirement is therefore:

$$((a * w) + (b * x) + (c * y) + (d * z))$$

A file system using a 1 KB block size (the smallest file system block size) allocates approximately eight times more memory at mount time than a file system of the same size using a 8 KB block size (the largest file system block size). For this reason, the Veritas Access file system defaults to a block size of 8 KB if a block size is not specified when creating a file system.

Some customers might like to create small file systems using a 1 KB file system block size and subsequently grow the file system size significantly, as the file system block size cannot be changed after the file system is created. This procedure can result in very large file systems using a 1 KB block size that can result in an unexpectedly large allocation of system memory at mount time.

A Clustered File System (CFS) primary mount requires slightly more memory allocated at mount-time than a CFS secondary. The performance team recommends that the memory utilization of a CFS primary be used as the guideline for calculating the file system mount-time memory requirement.

Table B-2 Memory footprint of 16 file systems with 32 TB size each - CFS primary mount

	32 TB each file system			
Block size/file system	CFS primary mount			
	Memory used (MB)			
	1 KB	2 KB	4 KB	8 KB
1	329	164	82	41
2	659	328	165	82
3	988	491	248	125
4	1326	657	337	166
5	1649	821	414	210

Table B-2 Memory footprint of 16 file systems with 32 TB size each - CFS primary mount (*continued*)

	32 TB each file system			
6	1977	985	498	249
7	2306	1150	581	291
8	2635	1329	665	333
9	2964	1483	747	375
10	3293	1646	829	418
11	3624	1810	913	459
12	3953	1975	995	534
13	4281	2140	1077	546
14	4614	2307	1161	589
15	4942	2471	1243	629
16	5272	2636	1325	671

Table B-3 Memory footprint of 16 file systems with 32 TB size each - CFS secondary mount

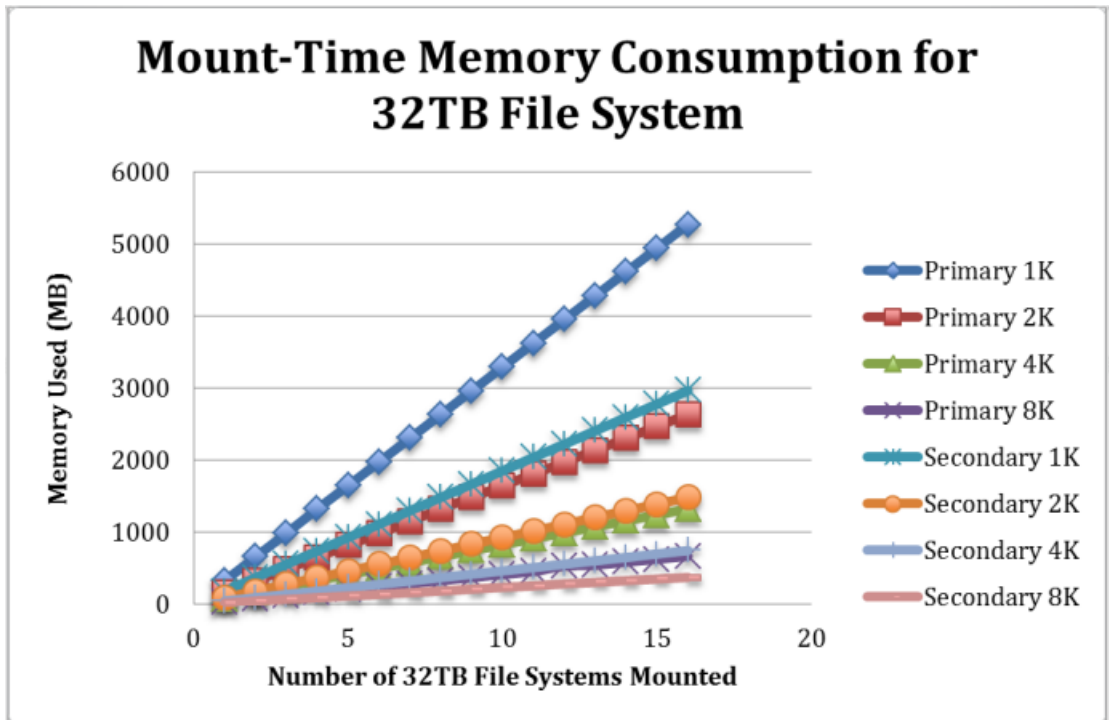
	32 TB each file system			
Block size/file system	CFS secondary mount			
	Memory used (MB)			
	1 KB	2 KB	4 KB	8 KB
1	187	93	47	21
2	372	186	94	48
3	558	279	139	71
4	742	371	186	94
5	929	465	233	117
6	1113	557	280	140
7	1300	650	326	164

Table B-3 Memory footprint of 16 file systems with 32 TB size each - CFS secondary mount (*continued*)

	32 TB each file system			
8	1485	743	373	187
9	1670	837	419	213
10	1854	928	465	237
11	2040	1020	512	259
12	2224	1114	558	286
13	2410	1208	606	306
14	2596	1301	652	330
15	2780	1393	701	353
16	2966	1485	747	376

Figure B-1 provides the guideline for the system memory utilization at mount time.

Figure B-1 Mount-time memory consumption for 32 TB file systems



Index

A

About

- cloud gateway 251
- compressed file format 342
- compressing files 341
- concurrent access 270
- configuring SmartIO 371
- FastResync 226
- file compression attributes 342
- file compression block size 343
- integration with OpenStack Cinder 303
- policies for scale-out file systems 258
- SNMP notifications 200

about

- Active Directory (AD) 122
- bonding Ethernet interfaces 37
- buckets and objects 192
- changing share properties 294
- configuring CIFS for AD domain mode 125
- configuring disks 67
- configuring routing tables 51
- configuring storage pools 68
- configuring the policy of each tiered file system 362
- configuring Veritas Access for CIFS 119
- configuring Veritas Access to use jumbo frames 46
- Continuous replication failover and fallback 431
- creating and maintaining file systems 210
- data deduplication 322
- episodic replication job failover and fallback 404
- Ethernet interfaces 42
- FTP 174
- FTP local user set 182
- FTP set 176
- I/O fencing 83
- IP load balancing 55
- iSCSI 87
- leaving AD domain 128
- managing CIFS shares 288
- Multi-protocol support for NFS with S3 194

about (*continued*)

- NFS file sharing 275
- scale-out file systems 211
- setting trusted domains 132
- shares 269
- snapshot schedules 441
- snapshots 435
- storage provisioning and management 67
- storing account information 145
- striping file systems 221
- the IP addresses for the Ethernet interfaces 42
- Veritas Access continuous replication 409
- Veritas Access episodic replication 381

About Configuring

- network 37

about maximum IOPS 228

about Veritas Access

- as an iSCSI target 98

accessing

- episodic replication destinations 404
- man pages 479
- the Veritas Access CLI 24
- Veritas Access product documentation 478

Active Directory

- setting the trusted domains for 144

Active Directory (AD)

- about 122
- joining Veritas Access to 124
- verifying Veritas Access has joined successfully 125

AD domain mode

- changing domain settings 129
- configuring CIFS 125
- security settings 129
- setting domain 126
- setting domain user 126
- setting security 126
- starting CIFS server 126

AD interface

- using 130

AD trusted domains

- disabling 144

- add local user
 - FTP 181
- adding
 - a column to a file system 236
 - a column to a tiered file system 358
 - a severity level to an email group 200
 - a syslog server 203
 - an email address to a group 200
 - an email group 200
 - CIFS share 294
 - external NetBackup master server to work with
 - Veritas Access 463
 - filter to a group 200
 - IP address to a cluster 43
 - Master, System Administrator, and Storage
 - Administrator users 32
 - mirror to a file system 234
 - mirror to a tier of a file system 359
 - mirrored tier to a file system 356
 - mirrored-striped tier to a file system 356
 - NetBackup media server 463
 - second tier to a file system 356
 - SNMP management server 207
 - striped tier to a file system 356
 - striped-mirror tier to a file system 356
 - users 32
 - VLAN interfaces 46
- adding a mapping
 - between CIFS and NFS users 153
- adding and configuring
 - Veritas Access to the Kerberos realm 116
- adding or deleting patterns
 - included in the list of files for backups 465
- aio_fork option
 - setting 162
- alerts
 - file system removing 247
- allowing
 - metadata information to be written on the
 - secondary tier 368
 - specified users and groups access to the CIFS
 - share 296
- as an iSCSI target
 - about Veritas Access 98
- Authenticate
 - NFS clients 115
- authenticating
 - NFS clients using Kerberos 115

- authentication
 - configuring the LDAP client using the CLI 143

B

- back up
 - Veritas Access file system over a SAN client 469
- backup
 - displaying excluded files 464
- backup or restore
 - NetBackup snapshot 470
- backup services
 - displaying the status of 467
 - starting 467
 - stopping 467
- Best practices
 - using compression 343
- best practices
 - creating file systems 217
 - for using the Veritas Access deduplication
 - feature 325
 - setting relocation policies 363
- bind distinguished name
 - setting for LDAP server 58
- bonding
 - Ethernet interfaces 38
- bonding Ethernet interfaces
 - about 37
- bringing
 - file system online or offline 233
- buckets and objects
 - about 192

C

- cache area
 - monitoring 377
 - verifying 377
 - viewing statistics 379
- Cache behavior
 - customizing 378
- cache object
 - destroying for an instant rollback 457
- cache objects
 - listing 455
- caching mode
 - setting 377
- caching VxFS in read mode 372
- caching VxFS in write mode 376

- changing
 - an IP address to online
 - on any running node 43
 - domain settings for AD domain mode 129
 - local CIFS user password 163
 - security settings 122
 - security settings after CIFS server is stopped 122
 - share properties about 294
- changing the caching mode 377
- checking
 - and repairing a file system 243
 - for stale mirrors on file systems 245
 - I/O fencing status 84
 - on the status of the NFS server 112
- CIFS
 - allowing specified users and groups access to the CIFS share 296
 - configuring schema extensions 139
 - denying specified users and groups access to the CIFS share 297
 - export options 290
 - mapuser commands 152
 - standalone mode 120
 - using multi-domain controller support 128
- CIFS aio_fork option
 - setting 162
- CIFS and NFS protocols
 - share directories 271
- CIFS clustering modes
 - about 119
- CIFS data migration
 - enabling 165
- CIFS home directories
 - quotas 75
- CIFS operating modes
 - about 119
- CIFS server
 - changing security settings after stopped 122
 - configuring with the LDAP backend 143
 - starting 149
 - trusted domains that are allowed access 132
- CIFS server status
 - standalone mode 121
- CIFS service
 - standalone mode 121
- CIFS share
 - adding 294
 - deleting 298
 - exporting as a directory 288
- CIFS share (*continued*)
 - exporting the same file system/directory as a different CIFS share 289
 - making shadow copy aware 300
 - modifying 299
- CIFS share and home directory
 - migrating from ctdb to normal clustering mode 161
- CIFS shares
 - scale-out file system 300
- CIFS shares and home directories
 - migrating from ctdb clustering modes 160
 - migrating from normal to ctdb clustering mode 161
- CIFS snapshot
 - exporting 298
- CIFS/NFS sharing
 - mapping user names 151
- clearing
 - DNS domain names 40
 - DNS name servers 40
 - LDAP configured settings 58
- CLI
 - configure and manage storage 77
- client configurations
 - displaying 61
 - LDAP server 61
- Cloud gateway
 - about 251
- cloud tier
 - obtaining data usage statistics 265
- Cluster
 - Excluding PCI IDs 50
- cluster
 - adding an IP address to 43
 - changing an IP address to online for any running node 43
 - displaying all the IP addresses for 43
- clustering modes
 - ctdb 159
- columns
 - adding or removing 236
 - adding or removing from a tiered file system 358
- command history
 - displaying 29
- Command-Line Interface (CLI)
 - getting help on how to use 27
- communicating
 - source and destination clusters 385, 417

- Compressed file format
 - about 342
- Compressing files
 - about 341
 - use cases 343
- compression
 - tasks 343
- Concurrent access
 - about 270
- concurrent access
 - object access service 274
- Configure and manage storage
 - CLI 77
- Configuring
 - Object Store server 188
- configuring
 - AD schema with CIFS-schema extensions 139
 - backups using NetBackup or other third-party
 - backup applications 468
 - CIFS for standalone mode 120
 - CIFS server with the LDAP backend 143
 - cloud gateway 252
 - deduplication 327
 - event notifications 198
 - IP routing 52
 - iSCSI devices 88
 - iSCSI discovery 89
 - iSCSI initiator 87
 - iSCSI initiator name 88
 - iSCSI targets 92
 - job resynchronization 403
 - NetBackup virtual IP address 466
 - NetBackup virtual name 466
 - NFS client for ID mapping 115
 - NIC devices 47
 - NSS lookup order 63
 - the policy of each tiered file system 362
 - Veritas Access for CIFS 119
 - VLAN interfaces 46
 - Windows Active Directory as an IDMAP
 - backend 138
- configuring CIFS share
 - secondary storage for an Enterprise Vault
 - store 289
- configuring disks
 - about 67
- configuring Ethernet interfaces
 - about 42
- configuring NetBackup
 - prerequisites 460
- configuring routing tables
 - about 51
- Configuring SmartIO
 - about 371
- configuring storage pools
 - about 68
- configuring the cloud as a tier
 - scale-out file systems 253
- Configuring the Object Store server
 - use case 188
- Configuring Veritas Access
 - ID mapping for NFS version 4 114
- Configuring Veritas Access for NetBackup
 - workflow 461
- configuring Veritas Access to use jumbo frames
 - about 46
- Considerations
 - creating a file system 217
- continuous replication
 - display status 424
 - setting up file system 421
 - unconfiguring 430
- Continuous replication failover and failback
 - about 431
- converting
 - existing file system into a cache object 453
- coordinator disks
 - replacing 84
- creating
 - full-sized rollback 450
 - local CIFS group 164
 - local CIFS user 163
 - mirrored file systems 229
 - mirrored-stripe file systems 229
 - OpenStack Manila file share 317
 - OpenStack Manila share snapshot 320
 - OpenStack Manila share type 316
 - RPO report 404
 - share backend on the OpenStack controller
 - node 315
 - shared cache object 453
 - simple file systems 229
 - snapshot schedules 442
 - snapshots 436
 - space-optimized instant rollbacks 450
 - storage pools 68
 - striped file systems 229

- creating *(continued)*
 - striped-mirror file systems 229
- Creating a file system
 - considerations 217
- creating and maintaining file systems
 - about 210
- creating and scheduling a policy
 - scale-out file system 261
- creating directory
 - FTP 175
- creating file systems
 - best practices 217
- ctdb clustering mode
 - about 159
 - directory-level share support 288
 - switching the clustering mode 160
- current Ethernet interfaces and states
 - displaying 43
- current users
 - displaying list 32
- customizing server options
 - FTP 179

D

- data deduplication
 - about 322
- data flow
 - continuous replication asynchronous mode 413
 - continuous replication synchronous mode 411
- decreasing
 - size of a file system 239
- deduplication
 - use-cases 323
- deduplication scenario
 - end-to-end 326
- deduplication workflow
 - overview 324
- default
 - passwords
 - resetting Master, System Administrator, and Storage Administrator users 32
- defining
 - what to replicate 395
- defrag 240
- defragmenting
 - file systems 240
- delete local user
 - FTP 181

- deleting
 - already configured SNMP management server 207
 - CIFS share 298
 - configured mail server 200
 - configured NetBackup media server 463
 - email address from a specified group 200
 - email groups 200
 - filter from a specified group 200
 - home directories 158
 - home directory of given user 158
 - local CIFS group 164
 - local CIFS user 163
 - NFS options 281
 - route entries from routing tables of nodes in cluster 52
 - severity from a specified group 200
 - snapshot schedules 445
 - syslog server 203
 - users 32
 - VLAN interfaces 46
- denying
 - specified users and groups access to the CIFS share 297
- description of Veritas Access continuous replication 410
- description of Veritas Access episodic replication 382
- destroying
 - a file system 248
 - cache object of an instant rollback 457
 - I/O fencing 84
 - instant rollbacks 453
 - snapshots 438
 - storage pools 68
- directories
 - displaying exported 280
 - unexporting the share 281
- directory-level share support
 - ctdb clustering mode 288
- disabling
 - AD trusted domains 144
 - creation of home directories 158
 - DNS settings 40
 - FastResync option 244
 - I/O fencing 84
 - LDAP clients
 - configurations 61
 - NetBackup SAN client 461
 - NIS clients 62

- disabling (*continued*)
 - NTLM 131
 - quota limits used by snapshots 438
 - disk
 - formatting 81
 - removing 82
 - disk layout versions
 - upgrading 248
 - Disk quotas
 - CIFS 71
 - file systems 71
 - usage 71
 - displaying
 - all the IP addresses for cluster 43
 - command history 29
 - current Ethernet interfaces and states 43
 - current list of SNMP management servers 207
 - DNS settings 40
 - events on the console 205
 - excluded files from backup 464
 - existing email groups or details 200
 - exported directories 280
 - file system alert values 247
 - file systems that can be exported 276
 - files moved or pruned by running a policy 367
 - home directory usage information 158
 - included files for backups 464
 - information for all disk devices for the nodes in a cluster 77
 - LDAP client configurations 61
 - LDAP configured settings 58
 - list of current users 32
 - list of SmartTier systems 361
 - list of syslog servers 203
 - local CIFS group 164
 - local CIFS user 163
 - NetBackup configurations 467
 - NFS statistics 113
 - NIS-related settings 62
 - NSS configuration 63
 - policy of each tiered file system 354
 - routing tables of the nodes in the cluster 52
 - schedules for tiered file systems 366
 - share properties 295
 - snapshot quotes 438
 - snapshot schedules 445
 - snapshots 437
 - snapshots that can be exported 276
 - status of backup services 467
 - displaying (*continued*)
 - tier location of a specified file 365
 - time interval or number of duplicate events for notifications 205
 - values of the configured SNMP notifications 207
 - values of the configured syslog server 203
 - VLAN interfaces 46
 - displaying a mapping
 - between CIFS and NFS users 153
 - displaying WWN information 79
 - DNS
 - domain names
 - clearing 40
 - name servers
 - clearing 40
 - specifying 40
 - settings
 - disabling 40
 - displaying 40
 - enabling 40
 - domain
 - setting 149
 - setting user name 149
 - domain controller
 - setting 149
 - domain name
 - for the DNS server
 - setting 40
- ## E
- email address
 - adding to a group 200
 - deleting from a specified group 200
 - email groups
 - adding 200
 - deleting 200
 - displaying existing and details 200
 - enabling
 - CIFS data migration 165
 - DNS settings 40
 - FastResync for a file system 243
 - I/O fencing 84
 - LDAP client configurations 61
 - NetBackup SAN client 461
 - NIS settings 62
 - NTLM 131
 - quota limits used by snapshots 438
 - enabling, disabling, and displaying
 - file system quotas 72

- encryption
 - KMS 215
- Enterprise Vault store
 - configuring CIFS share as secondary storage 289
- episodic replication destination file system behavior
 - about 403
- episodic replication destinations
 - accessing 404
- episodic replication job
 - displaying status 402
 - enabling compression 401
 - managing 397
 - show job 402
- episodic replication jobs
 - maximum number of parallel 397
- episodic replication unit
 - setting up files to exclude 391
- erasure-coding
 - NFS use case 214
- Ethernet interfaces
 - bonding 38
- event notifications
 - configuring 198
 - displaying time interval for 205
- event reporting
 - setting events for 205
- events
 - displaying on the console 205
- excluded files
 - displaying 464
- excluding directories and files
 - setting up 391
- Excluding PCI IDs
 - cluster 50
- export options
 - CIFS 290
- exporting
 - an NFS share 276
 - CIFS snapshot 298
 - directory as a CIFS share 288
 - events in syslog format to a given URL 204
 - NFS snapshot 286
 - same file system/directory as a different CIFS share 289
 - SNMP MIB file to a given URL 207
- exporting for Kerberos authentication
 - NFS share 283

F

- failover 332
- failover and fallback
 - about 431
 - about episodic replication 404
- FastResync
 - about 226
- File compression attributes
 - about 342
- File compression block size
 - about 343
- file sharing
 - using FTP 185
- file system
 - converting into a cache object 453
- file system alert values
 - displaying 247
- file system alerts
 - removing 247
 - setting 246
- File system mount-time
 - memory usage 481
- file system quotas
 - enabling, disabling, and displaying 72
 - setting and displaying 73
- file systems
 - adding a mirror to 234
 - bringing online or offline 233
 - checking and repairing 243
 - checking for stale mirrors 245
 - creating 229
 - decreasing the size of 239
 - defragmenting 240
 - destroying 248
 - disabling FastResync option 244
 - enabling FastResync 243
 - increasing the size of 237
 - listing with associated information 234
 - removing a mirror from 234
 - restoring from an instant rollback 451
 - SmartTier
 - displaying 361
 - that can be exported
 - displayed 276
 - types of layout 219
- filter
 - about 199
 - adding to a group 200
 - deleting from a specified group 200

- firewall setting 54
- forcefully
 - importing new LUNs for new or existing pools 80
- formatting
 - a disk 81
- FTP
 - about 174
 - add local user 181
 - creating directory 175
 - customizing server options 179
 - delete local user 181
 - local user password 181
 - local user set download bandwidth 183
 - local user set home directory 183
 - local user set maximum connections 183
 - local user set maximum disk usage 183
 - local user set maximum files 183
 - local user set upload bandwidth 183
 - logupload 180
 - server start 175
 - server status 175
 - server stop 175
 - session show 180
 - session showdetail 180
 - session terminate 180
 - show local users 181
- FTP local user set
 - about 182
- FTP set
 - about 176

G

- group membership
 - managing 163

H

- hiding
 - system files when adding a CIFS normal share 295
- history command
 - using 29
- home directories
 - setting up 156
- home directory file systems
 - setting 155
- home directory of given user
 - deleting 158

- home directory usage information
 - displaying 158
- hostname or IP address
 - setting for LDAP server 58
- how to use
 - Command-Line Interface (CLI) 27

I

- I/O fencing
 - about 83
 - checking status 84
 - destroying 84
 - disabling 84
 - enabling 84
- ID mapping for NFS version 4
 - configuring Veritas Access 114
- importing
 - new LUNs forcefully for new or existing pools 80
- included files
 - for backups 464
- increasing
 - LUN storage capacity 81
 - size of a file system 237
- initiating host discovery of LUNs 80
- installing
 - NetBackup client 470
- instant rollbacks
 - about 448
 - bringing online 452
 - creating a shared cache object 453
 - creating full-sized 450
 - creating space-optimized 450
 - destroying 453
 - listing 451
 - refreshing from a file system 452
 - restoring a file system from 451
 - taking offline 452
- integration of Veritas Access
 - with OpenStack 302
- Integration with OpenStack Cinder
 - about 303
- integration with Veritas Access
 - NetBackup 459
- IP addresses
 - adding to a cluster 43
 - displaying for the cluster 43
 - modifying 43
 - removing from the cluster 43

IP addresses for the Ethernet interfaces

about 42

IP load balancing 55

IP routing

configuring 52

iSCSI

about 87

iSCSI devices

configuring 88

iSCSI discovery

configuring 89

iSCSI initiator

configuring 87

iSCSI initiator name

configuring 88

iSCSI target service

managing 101

iSCSI targets

configuring 92

managing 99

J

job resynchronization

configuring 403

joining

Veritas Access to Active Directory (AD) 124

K

Kerberos authentication

authenticating NFS clients 115

Kerberos realm

adding and configuring Veritas Access for 116

Kerberos share

mounting from the NFS client 284

Kernel-based

NFS server 109

L

layouts

types of file system 219

LDAP

before configuring 57

LDAP client

configuring for authentication using the CLI 143

LDAP password hash algorithm

setting password for 58

LDAP server

clearing configured settings 58

LDAP server *(continued)*

disabling client configurations 61

displaying client configurations 61

displaying configured settings 58

enabling client configurations 61

setting over SSL 58

setting port number 58

setting the base distinguished name 58

setting the bind distinguished name 58

setting the hostname or IP address 58

setting the password hash algorithm 58

setting the root bind DN 58

setting the users, groups, and netgroups base
DN 58

leaving

AD domain 128

list of SmartTier file systems

displaying 361

listing

all file systems and associated information 234

all of the files on the specified tier 361

cache objects 455

free space for storage pools 68

instant rollbacks 451

storage pools 68

Loading files to the cache 378

local CIFS groups

creating 164

deleting 164

displaying 164

local CIFS user

creating 163

deleting 163

displaying 163

local CIFS user password

changing 163

local user and groups

managing 163

local user password

FTP 181

local user set download bandwidth

FTP 183

local user set home directory

FTP 183

local user set maximum connections

FTP 183

local user set maximum disk usage

FTP 183

- local user set maximum files
 - FTP 183
- local user set upload bandwidth
 - FTP 183
- logupload
 - FTP 180
- LUN storage capacity
 - increasing 81
- LUNs
 - initiating host discovery 80
 - managing 102

M

- mail server
 - deleting the configured mail server 200
 - obtaining details for 200
 - setting the details of external 200
- man pages
 - how to access 479
- managing
 - CIFS shares 288
 - continuous replication 422
 - group membership 163
 - home directories 154
 - iSCSI target service 101
 - iSCSI targets 99
 - local users and groups 163
 - LUNs 102
 - mappings with iSCSI initiators 105
 - users 106
- managing NFS shares
 - using netgroups 281
- mapping
 - of UNIX users from LDAP to Windows users 154
- mappings with iSCSI initiators
 - managing 105
- mapuser commands
 - about 152
- Master, System Administrator, and Storage
 - Administrator users
 - adding 32
- maximum number
 - parallel episodic replication jobs 397
- Memory usage
 - file system mount-time 481
- metadata information
 - allowing to be written on the secondary tier 368
 - restricting to the primary tier only 368

- migrating
 - CIFS share and home directory from ctdb to
 - normal clustering mode 161
 - CIFS shares and home directories 160
 - CIFS shares and home directories from normal
 - to ctdb clustering mode 161
- mirrored file systems
 - creating 229
- mirrored tier
 - adding to a file system 356
- mirrored-stripe file systems
 - creating 229
- mirrored-striped tier
 - adding to a file system 356
- modifying
 - an IP address 43
 - CIFS share 299
 - policy of a tiered file system 354
 - schedule of a tiered file system 366
 - snapshot schedules 444
 - tunables for iSCSI 95
- more command
 - using 29
- mounting
 - NFS share from the NFS client 284
- mounting snapshots 440
- moving files between tiers
 - scale-out file system 255
- Multi-protocol support for NFS with S3
 - limitations 194

N

- naming requirement for new users 34
- navigating CLI
 - Veritas Access 24
- NetBackup
 - configuring 468
 - configuring NetBackup virtual IP address 466
 - configuring virtual name 466
 - displaying configurations 467
 - integration with Veritas Access 459
 - Snapshot Client 460
 - snapshot methods 460
- NetBackup master server
 - configuring to work with Veritas Access 463
- NetBackup media server
 - adding 463
 - deleting 463

- NetBackup SAN client
 - disabling 461
 - enabling 461
- NetBackup snapshot
 - backup or restore 470
- network interfaces
 - swapping 49
- NFS client
 - configuring for IP mapping 115
- NFS clients
 - authenticating 115
- NFS file sharing
 - about 275
- NFS options
 - deleting 281
- NFS server
 - about 108
 - checking on the status 112
 - kernel-based 109
 - starting 112
 - stopping 112
- NFS Servers
 - switching 109
- NFS share
 - exporting 276
 - exporting for Kerberos authentication 283
- NFS shares
 - managing using netgroups 281
- NFS snapshot
 - exporting 286
- NFS statistics
 - displaying 113
 - resetting 113
- NFS-Ganesha server
 - use 109
- NFS-Ganesha version 3 and version 4
 - recommended tuning parameters 110
- NIC devices
 - configuring 47
- NIS
 - clients
 - disabling 62
 - enabling 62
 - domain name
 - setting on all the nodes of cluster 62
 - related settings
 - displaying 62
 - server name
 - setting on all the nodes of cluster 62

- nocache caching mode 377
- node
 - in a cluster
 - displaying information for all disk devices 77
- node or storage connection failures
 - when using Oracle Direct NFS 170
- NSS
 - displaying configuration 63
 - lookup order
 - configuring 63
- NTLM
 - disabling 131
 - enabling 131

O

- object server 187
- Object Store server
 - configuring 188
- objectstore buckets 194
- obtaining
 - details of the configured email server 200
- offline
 - taking an instant rollback offline 452
- online
 - bringing an instant rollback online 452
- OpenStack
 - about the integration with OpenStack 302
- OpenStack Manila
 - integration with Veritas Access 312
- OpenStack Manila file share
 - creating 317
- OpenStack Manila share snapshot
 - creating 320
- OpenStack Manila share type
 - creating 316
- Oracle database performance 167
- Oracle Direct NFS 166
 - architecture 169
 - node or storage connection failures 170
- overview
 - deduplication workflow 324

P

- password
 - changing a user's password 32
- patterns
 - adding or deleting from the list of files included
 - in backups 465

- physical and logical data on a file system 323
- Pinning files to the cache 378
- planned failback
 - process overview 407, 433
- planned failover
 - process overview 406, 432
- policies
 - about 362
 - displaying files moved or pruned by running 367
 - displaying for each tiered file system 354
 - modifying for a tiered file system 354
 - relocating from a tiered file system 364
 - removing from a tiered file system 354
 - running for a tiered file system 354
- Policies for scale-out file systems
 - about 258
- prerequisites
 - configuring NetBackup 460
- preserving
 - snapshot schedules 445
- Private network
 - configure 37
- privileges
 - about 34
- Public network
 - configure 37

Q

- quota commands
 - setting and displaying file system quotas 73
- quota limits
 - enabling or disabling snapshot 438
- quotas
 - CIFS home directories 75
 - setting user quotas for users of specified groups 75

R

- read caching 371
- read caching mode 377
- recommended tuning parameters
 - NFS-Ganesha version 3 and version 4 110
- refreshing
 - instant rollbacks from a file system 452
- registering
 - NetBackup master server 463
- Relationship between
 - physical and logical data on a file system 323

- relocating
 - policy of a tiered file system 364
- removing
 - a column from a file system 236
 - a column from a tiered file system 358
 - a disk 82
 - deduplication 338
 - IP address from the cluster 43
 - mirror from a file system 234
 - mirror from a tier spanning a specified disk 359
 - mirror from a tier spanning a specified pool 359
 - mirror from a tiered file system 359
 - policy of a tiered file system 354
 - schedule of a tiered file system 366
 - snapshot schedules 445
 - tier from a file system 368
- removing a mapping
 - between CIFS and NFS users 153
- renaming
 - storage pools 68
- replacing
 - coordinator disks 84
- replicating file systems
 - setting up 389
- resetting
 - default passwords
 - Master, System Administrator, and Storage Administrator users 32
 - NetBackup master server 463
 - NFS statistics 113
 - virtual IP address of NetBackup 466
- restoring
 - a file system from an instant rollback 451
 - snapshots 441
- restricting
 - metadata information to the primary tier only 368
- resynchronizing
 - stale mirrors on file systems 245
- roles
 - about 34
- route entries
 - deleting from routing tables 52
- routing tables
 - of the nodes in the cluster
 - displaying 52
- RPO report
 - creating 404
- running
 - policy of a tiered file system 354

running manually
deduplication 331

S

SAN client
back up 469

scale-out file system
CIFS shares 300
creating and scheduling a policy 261
file sharing 185
move the files 255

scale-out file systems
about 211
configuring the cloud as a tier 253
data usage in the cloud tier 265

scale-out fsck
about 226

schedule
displaying for tiered file systems 366
modifying for a tiered file system 366
removing from a tiered file system 366

scheduling
deduplication 333
episodic replication 393

second tier
adding to a file system 356

security
standalone mode 121

security settings
AD domain mode 129
changing 122

server start
FTP 175

server status
FTP 175

server stop
FTP 175

session show
FTP 180

session showdetail
FTP 180

session terminate
FTP 180

setting
AD domain mode 126
aio_fork option 162
base distinguished name for the LDAP server 58
bind distinguished name for LDAP server 58
details of the external mail server 200

setting (*continued*)
domain 149
domain controller 149
domain name for the DNS server 40
domain user name 149
events for event reporting 205
file system alerts 246
filter of the syslog server 203
home directory file systems 155
IDMAP backend to ad for access to CIFS 137
IDMAP backend to hash for accessing CIFS 136
IDMAP backend to ldap for trusted domain access to CIFS 135
IDMAP backend to rid for access to CIFS 133
LDAP password hash algorithm 58
LDAP server hostname or IP address 58
LDAP server over SSL 58
LDAP server port number 58
LDAP users, groups, and netgroups base DN 58
NIS domain name on all the nodes of cluster 62
prior to configuring LDAP 57
retention 227
root bind DN for the LDAP server 58
severity of the syslog server 203
SNMP filter notifications 207
SNMP severity notifications 207
the NIS server name on all the nodes of cluster 62
trusted domains 132
trusted domains for the Active Directory 144
user quotas for users of specified groups 75
WORM over NFS 228

setting domain user
AD domain mode 126

setting parameters
deduplication 336

setting relocation policies
best practices 363

setting security
AD domain mode 126

setting the caching mode 377

setting up
deduplication 326
home directories 156
replicating file systems 389

setting up an episodic replication unit
to exclude directories and files 391

severity levels
about 199

- severity levels *(continued)*
 - adding to an email group 200
- severity notifications
 - setting 207
- shadow copy
 - making a CIFS share aware 300
- share backend
 - creating on the OpenStack controller node 315
- share directories
 - CIFS and NFS protocols 271
- share properties
 - displaying 295
- shared cache object
 - creating 453
- shares
 - about 269
- show local users
 - FTP 181
- showing
 - snapshot schedules 445
- SmartIO
 - about 370
- snapshot methods
 - NetBackup 460
- snapshot schedules
 - about 441
 - creating 442
 - deleting 445
 - displaying 445
 - modifying 444
 - preserving 445
 - removing 445
 - showing 445
- snapshots
 - about 435
 - creating 436
 - destroying 438
 - displaying 437
 - displaying quotas 438
 - enabling or disabling quota limits 438
 - mounting 440
 - restoring 441
 - that can be exported
 - displayed 276
 - unmounting 440
- SNMP
 - filter notifications
 - setting 207
 - SNMP *(continued)*
 - management server
 - adding 207
 - deleting configured 207
 - displaying current list of 207
 - MIB file
 - exporting to a given URL 207
 - notifications
 - displaying the values of 207
 - server
 - setting severity notifications 207
- SNMP notifications
 - about 200
- solid-state drives (SSDs)
 - about 370
- source and destination clusters
 - communicating 385, 417
- specific workload
 - creating a tuned file system 224
- specified group
 - deleting a severity from 200
- specifying
 - DNS name servers 40
- SSL
 - setting the LDAP server for 58
- standalone mode
 - CIFS server status 121
 - CIFS service 121
 - security 121
- starting
 - backup services 467
 - CIFS server 149
 - NFS server 112
 - Veritas Access continuous replication 415
 - Veritas Access episodic replication 383
- starting CIFS server
 - AD domain mode 126
- statistics
 - data usage in the cloud tier 265
- stopping
 - backup services 467
 - NFS server 112
- storage pools
 - creating 68
 - destroying 68
 - listing 68
 - listing free space 68
 - renaming 68

- storage provisioning and management
 - about 67
- storing
 - account information 145
 - user and group accounts in LDAP 147
 - user and group accounts locally 147
- striped file systems
 - creating 229
- striped tier
 - adding to a file system 356
- striped-mirror file systems
 - creating 229
- striped-mirror tier
 - adding to a file system 356
- striping file systems
 - about 221
- swapping
 - network interfaces 49
- Switch
 - NFS servers 109
- switching
 - ctdb clustering mode 160
- syslog format
 - exporting events to a given URL 204
- syslog server
 - adding 203
 - deleting 203
 - displaying the list of 203
 - displaying the values of 203
 - setting the filter of 203
 - setting the severity of 203
- system files
 - hiding when adding a CIFS normal share 295

T

- tier
 - adding a tier to a file system 359
 - displaying location of a specified file 365
 - listing all of the specified files on 361
 - removing a mirror from 359
 - removing a mirror spanning a specified pool 359
 - removing from a file system 368
 - removing from a tier spanning a specified disk 359
- trusted domains
 - allowing access to CIFS when setting an IDMAP backend to ad 137
 - allowing access to CIFS when setting an IDMAP backend to hash 136

- trusted domains *(continued)*
 - allowing access to CIFS when setting an IDMAP backend to ldap 135
 - allowing access to CIFS when setting an IDMAP backend to rid 133
 - specifying which are allowed access to the CIFS server 132
- tunables for iSCSI
 - modifying 95
- tuned file system
 - creating for a specific workload 224

U

- unexporting
 - share of exported directory 281
- uninstalling
 - NetBackup client 470
- UNIX users from LDAP to Windows users
 - automatic mapping of 154
- unmounting snapshots 440
- Unpinning files from the cache 378
- unplanned fallback
 - process overview 408, 433
- unplanned failover
 - process overview 407, 433
- upgrading
 - disk layout versions 248
- Use
 - NFS-Ganesha server 109
- Use case
 - configuring the Object Store server 188
- Use cases
 - compressing files 343
- use-cases
 - deduplication 323
- user and group accounts in LDAP
 - storing 147
- user and group accounts locally
 - storing 147
- user names
 - mapping for CIFS/NFS sharing 151
- user roles and privileges
 - about 34
- users
 - adding 32
 - changing passwords 32
 - deleting 32
 - managing 106

- using
 - AD interface 130
 - history command 29
 - more command 29
 - multi-domain controller support in CIFS 128

- Using compression
 - best practices 343

- Using NFS Server 108

V

- verifying
 - deduplication 339
 - Veritas Access has joined Active Directory (AD) 125
- Veritas Access
 - about 18
 - accessing the CLI 24
 - integration with OpenStack Manila 312
 - key features 18
 - product documentation 478
- Veritas Access continuous replication
 - about 409
 - description of feature 410
 - starting 415
- Veritas Access deduplication feature
 - best practices for using 325
- Veritas Access episodic replication
 - about 381
 - compression 401
 - description of feature 382
 - scheduling 393
 - starting 383
- Veritas Access file systems
 - read caching 371
 - writeback caching 373
- Veritas Access SmartTier
 - about 351
- Veritas Access to the Kerberos realm
 - adding and configuring 116
- Veritas File System (VxFS)
 - configuring read caching 372
 - customizing caching 378
 - write caching 376
- viewing detailed cache statistics 379
- viewing file system cache statistics 379
- virtual IP address
 - configuring or changing for NetBackup 466
- virtual name
 - configuring for NetBackup 466

VLAN

- adding interfaces 46
- configuring interfaces 46
- deleting interfaces 46
- displaying interfaces 46

W

- what to replicate
 - defining 395
- Windows Active Directory
 - configuring as an IDMAP backend 138
- Workflow
 - configuring Veritas Access for NetBackup 461
- workflow
 - object server 187
- WORM over NFS
 - setting 228
- write caching 376
- writeback caching 373
 - flushing dirty data 375
 - tuning 374
- writeback caching mode 377
- WWN information
 - displaying 79