ARX® CLI Storage-Management Guide

810-0044-00

Publication Date

This manual was published on May 13, 2013.

Legal Notices

Copyright

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

Access Policy Manager, Advanced Client Authentication, Advanced Routing, APM, Application Security Manager, ARX, AskF5, ASM, BIG-IP, BIG-IQ, Cloud Extender, CloudFucious, Cloud Manager, Clustered Multiprocessing, CMP, COHESION, Data Manager, DevCentral, DevCentral [DESIGN], DNS Express, DSC, DSI, Edge Client, Edge Gateway, Edge Portal, ELEVATE, EM, Enterprise Manager, ENGAGE, F5, F5 [DESIGN], F5 Management Pack, F5 Networks, F5 World, Fast Application Proxy, Fast Cache, FirePass, Global Traffic Manager, GTM, GUARDIAN, IBR, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iApps, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, L7 Rate Shaping, LC, Link Controller, Local Traffic Manager, LTM, Message Security Manager, MSM, OneConnect, OpenBloX, OpenBloX [DESIGN], Packet Velocity, Policy Enforcement Manager, PEM, Protocol Security Manager, PSM, Real Traffic Policy Builder, Rosetta Diameter Gateway, ScaleN, Signaling Delivery Controller, SDC, SSL Acceleration, StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TMOS, Traffic Management Operating System, Traffix Diameter Load Balancer, Traffix Systems, Traffix Systems (DESIGN), Transparent Data Reduction, UNITY, VAULT, VIPRION, vCMP, virtual Clustered Multiprocessing, WA, WAN Optimization Manager, WebAccelerator, WOM, and ZoneRunner, are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by U.S. Patents 7,877,511; 7,958,347. This list is believed to be current as of May 13, 2013.

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software from several third-party vendors. Each vendor is listed below with the applicable copyright.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

Unless otherwise noted, the companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Revision History

September 2006 - Rev A

October 2006 - Rev B, updates for Software Release 2.4.2

January 2007 - Rev C, updates for Software Release 2.4.3

March 2007 - Rev D, updates for Software Release 2.5.0

May 2007 - Rev E, updates for Software Release 2.5.1

August 2007 - Rev F, add forest-to-forest trusts for Software Release 2.6.0

November 2007 - Rev G, minor changes in “show namespace,” Software Release 2.7.0

December 2007 - Rev H, updates for Software Release 3.0.0

February 2008 - Rev J, minor changes in “show” commands, Software Release 2.7.1

March 2008 - Rev K, convert to F5 format, Software Release 3.1.0

June 2008 - Rev L, updates for Software Release 3.2.0

June 2008 - Rev M, updates for Software Release 4.0.0

July 2008 - Rev N, add EMC-Mixed-Mode support for Software Release 4.0.1

October 2008 - Rev Q, document new CIFS features (ABE, rekey) in Software Release 4.1.0

June 2009 - Rev R, updates for Software Release 5.00.000

July 2009 - Rev S, minor updates for Software Release 5.00.001

July 2009 - Rev T, re-orient policy discussion toward latest best-practices, for Software Release 5.00.005

November 2009 - Rev U, updates for Software Release 5.01.000

July 2010 - Rev V, updates for Software Release 5.02.000

June 2011 - Rev W, updates for Software Release 6.00.000

September 2011 - Rev X, updates for Software Release 6.01.000
January 2012 - Rev Y, minor updates for 6.01.001 hot fix
July 2012 - Rev Z, updates for Software Release 6.02.000
October 2012 - Rev AA, updates for Software Release 6.03.000
June 2013 - Rev AB, updates for Software Release 6.04.000

Table of Contents

Introduction

Product Overview

Preparing for CIFS Authentication

Preparing for NFS Authentication

Examining Filers

Adding an External Filer

Configuring a Namespace

Adding a Direct Volume

Adding a Managed Volume

Configuring a Global Server

Configuring Front-End Services

Creating a Policy Schedule

Grouping Files Into Filesets

Migrating Filesets and Tiering

1

Introduction

- Overview
- The ARX
- Audience for this Manual
- Using this Manual
- Document Conventions
- CLI Overview
- Getting Started
- Sample Network
- Contacting Customer Service

Overview

This manual contains instructions and best practices for setting up and managing storage on the Adaptive Resource Switch (ARX®). These instructions focus on the Command-Line Interface (CLI).

Use this book after the ARX® is installed and connected to its clients and servers through IP. The platform's *Hardware Installation* manual explains how to install the ARX. You can set up basic networking through a GUI wizard, or work with the more-advanced CLI features described in the *ARX® CLI Network-Management Guide*.

The ARX

The Adaptive Resource Switch (ARX®) is a highly available and scalable solution that brings resource awareness to a file storage infrastructure, and adapts these resources to meet the demands of users and applications in real time. The ARX provides a file-virtualization layer that aggregates the total capacity and performance of your file storage. A *namespace* provides location-independent, transparent mapping of user requests onto the appropriate storage resource. You can configure policies that the switch enforces for the placement, replication and migration of files. Through policy configuration, the ARX adapts to the real-time demands of users and applications. The ARX thereby serves as a *resource proxy* for the files and services behind it.

Back-end Storage and Servers

The Adaptive Resource Switch aggregates heterogeneous file systems and storage into a unified pool of file storage resources. Through this unification, you can manage these resources to adapt to user demands and client applications. File storage assets can be differentiated based on user-defined attributes, enabling a class-of-storage model. You can reclaim stranded capacity through policy implementation for more effective storage utilization, and you can add capacity without disruption. Back-end resources are monitored for availability and performance, as well as user-access patterns that drive policy decisions.

Front-end Services

The Adaptive Resource Switch acts as an in-band file proxy for the Network File System (NFS) and Microsoft's Common Internet File System (CIFS) protocols.

Front-end services provide the file virtualization layer that masks the physical file storage from the user and application. The switch becomes the file access point, as opposed to the actual physical resource, providing file access through a namespace. Users and applications maintain a single consistent file path that is transparently mapped to the proper physical resource where the information resides.

Policy

The Adaptive Resource Switch provides policy-based resource switching. Through *policy* configuration, you can optimize the placement of files onto the appropriate storage resources and automatically adapt these resources based on user and application demand. The ARX performs file replication and migration based on performance, usage or other life-cycle characteristics, enabling you to implement a flexible file services strategy. Examples of policies include: migrating files to reclaim stranded capacity; migrating files across different tiers of storage based on access patterns and/or value; and replicating frequently accessed files for performance. The result is more efficient utilization and greater flexibility in file storage management.

Resilient Overlay Network (RON)

You can connect multiple ARXes with a Resilient Overlay Network (RON), which can reside on top of any IP network. This provides a network for distributing and accessing file storage. ARXes can replicate storage to other switches in the same RON, updating the replicas periodically as the writable master files change. This is called a *shadow copy*, where a source volume on one switch periodically copies its files to one or more *shadow volumes* on other switches. Clients can access the shadow volumes at multiple geographic locations, independent to where the source volume resides.

Audience for this Manual

This manual is intended for

- network technicians responsible for layer 1 and 2 networks,
- network engineers responsible for the Internet Protocol (IP) layer (layer 3),
- storage engineers who design and manage storage systems (SANs, NASes, and DASes), and
- crypto officers who manage all of the Critical Security Parameters (CSPs) of a network.

The text presumes that all readers are comfortable with a command-line interface (CLI), especially one based on the Cisco IOS.

Using this Manual

The next chapter shows some of the problems in today's storage networks and provides high-level instructions for using the ARX to solve those problems.

The remaining chapters are presented in the same order that you would use to configure storage on a new ARX. Before you begin, you must follow the instructions in your *Hardware Installation Guide* to install the switch, set up its management IP, and prepare it for CLI provisioning. A network engineer can then use ARX Manager or the *ARX® CLI Network-Management Guide* to set up the required networking parameters. Then you can follow the order of the chapters in this manual to

1. (for CIFS installations) configure some Windows security parameters referenced in other parts of the configuration,
2. (for NFS installations) add NIS netgroups and/or NFS access lists,
3. examine back-end filers (external NAS devices, or file servers with DAS) to determine their eligibility for use in a namespace,
4. add one or more back-end filers,
5. aggregate the filer storage into one or namespace volumes,
6. configure a global server and virtual server,
7. configure front-end services, such as NFS and CIFS, that clients can use to access the volume(s) through the global/virtual server,
8. configure namespace policy (capacity balancing, fileset configuration, and fileset-placement policy), and
9. configure a read-only shadow volume for a valued managed volume.

You can later return to any chapter to update the configuration at a particular layer.

Document Conventions

This manual uses the following conventions:

`this font` represents screen input and output;

- **`bold text`** represents input, and
- *`italic text`* appears for variable input or output.

this font is used for command-syntax definitions, which use the same rules for bold and italic.

Command-syntax definitions also use the following symbols:

- [*optional-argument*] - square brackets ([ ]) surround optional arguments;
- *choice1* | *choice2* - the vertical bar (|) separates argument choices;
- {*choice1* | *choice2* | *choice3*} - curly braces ({ }) surround a required choice;
- [*choice1* | *choice2*]* - an asterisk (*) means that you can choose none of them, or as many as desired (for example, “choice1 choice2” chooses both);
- {*choice1* | *choice2*}+ - a plus sign (+) means that you must choose one or more.

CLI Overview

The Command-Line Interface (CLI) has its commands grouped into modes. Modes are structured as a tree with a single root, *exec* mode. This section summarizes the mode structure and explains some CLI conventions.

Exec Mode

When you log into the CLI, you begin in exec mode. If the hostname is “bstnA,” the prompt appears as shown below:

```
bstnA>
```

You can access all global commands (such as show commands) from exec mode, and you can use the enable command to enter priv-exec mode.

```
bstnA> enable
```

Global Commands

You can access global commands from any mode, not just exec. Global commands include all show commands and terminal-control commands.

Priv-exec Mode

Priv-exec mode has the following prompt:

```
bstnA#
```

Priv-exec mode contains chassis-management commands, clock commands, and other commands that require privileges but do not change the network or storage configuration.

Priv-exec has two sub modes, cfg and gbl.

Cfg Mode

To enter cfg mode, use the config command:

```
bstnA# config
bstnA(cfg)#
```

Config mode contains all modes and commands for changing the configuration of the local switch, such as network configuration.

Gbl Mode

To enter gbl mode, use the global command:

```
bstnA# global
bstnA(gbl)#
```

Gbl mode controls all parameters that are shared in a redundant pair, such as namespaces and global servers.

Exiting a Mode

From any mode, use the exit command to return to its parent mode. From priv-exec mode, this command exits the CLI; to go from priv-exec mode back to exec mode, use the no enable command.

From any submode of cfg or gbl mode, you can return immediately to priv-exec mode by using the end command or pressing <Ctrl-z>.

Prompts

Prompts contain information about your position in the mode hierarchy as well as the name of the object you are configuring. For example, suppose you use the following command in gbl mode:

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])#
```

This command places you into a new mode, as indicated by the new CLI prompt. The prompt shows the name of the mode, "gbl-ns," and the name of the configuration object, a namespace called "wwmed." Abbreviations are used for mode names (for example, "ns" instead of "namespace") to conserve space on the command line.

When you descend to lower modes in the config tree, the prompt offers more information. To extend the previous example, suppose you enter the following command to configure the "/local" volume in the wwmed namespace:

```
bstnA(gbl-ns[wwmed])# volume /local
bstnA(gbl-ns-vol[wwmed~/local])#
```

The tilde character (~) separates a parent object from its child: "wwmed~/local" shows that you are in the "/local" volume under the "wwmed" namespace.

The no Convention

Most config commands have the option to use the "no" keyword to negate the command. For commands that create an object, the no form removes the object. For commands that change a default setting, the no form reverts back to the default. As an example,

```
bstnA(gbl-ns[wwmed])# no volume /local
```

removes the "/local" volume from the "wwmed" namespace.

The enable/no enable Convention

Many objects and configurations require you to enable them using the enable command before they can take effect. Likewise, many objects and configurations require you to first disable them using the no enable command before you can complete a related command or function. The no enable command does not remove an object; it only disables it until you re-enable it. The enable/no enable commands exist in many modes and submodes in the CLI.

For example, the following command sequence enables the namespace named “wwmed:”

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# enable
bstnA(gbl-ns[wwmed])# ...
```

Getting Started

For the initial login, refer to the instructions for booting and configuring the switch in the appropriate *Hardware Installation Guide*.

For subsequent logins, use the following steps to log into the F5 CLI:

1. If you are on-site, you can connect a serial line to the serial console port. This port is labeled ‘Console’ or ‘10101’ (depending on your ARX platform). By default, the port is set for 9600 baud, 8, N, 1.

   You can also telnet to the switch’s management interface. For example:

   telnet 10.10.10.10

   In either case, a login prompt appears:

```
Username:
```

2. Enter your username and password. For example:

```
Username: admin
Password: acopia
```

   The CLI prompt appears:

```
SWITCH>
```

   The name, “SWITCH,” is the default hostname. The hostname is reset as part of the initial-boot process, so it is likely that yours will differ.

Entering Cfg or Gbl Mode

As discussed above, the CLI contains two major configuration modes: cfg and gbl. The cfg mode contains submodes for configuring locally-scoped parameters, only applicable to the local ARX. These parameters include layer-2, layer-3, and chassis configuration. Gbl mode applies to configuration that is shared among both switches in a redundant pair, such as namespaces and global servers.

After you log into the CLI, use the config command to enter cfg mode:

```
SWITCH> enable
SWITCH# configure
SWITCH(cfg)#
```

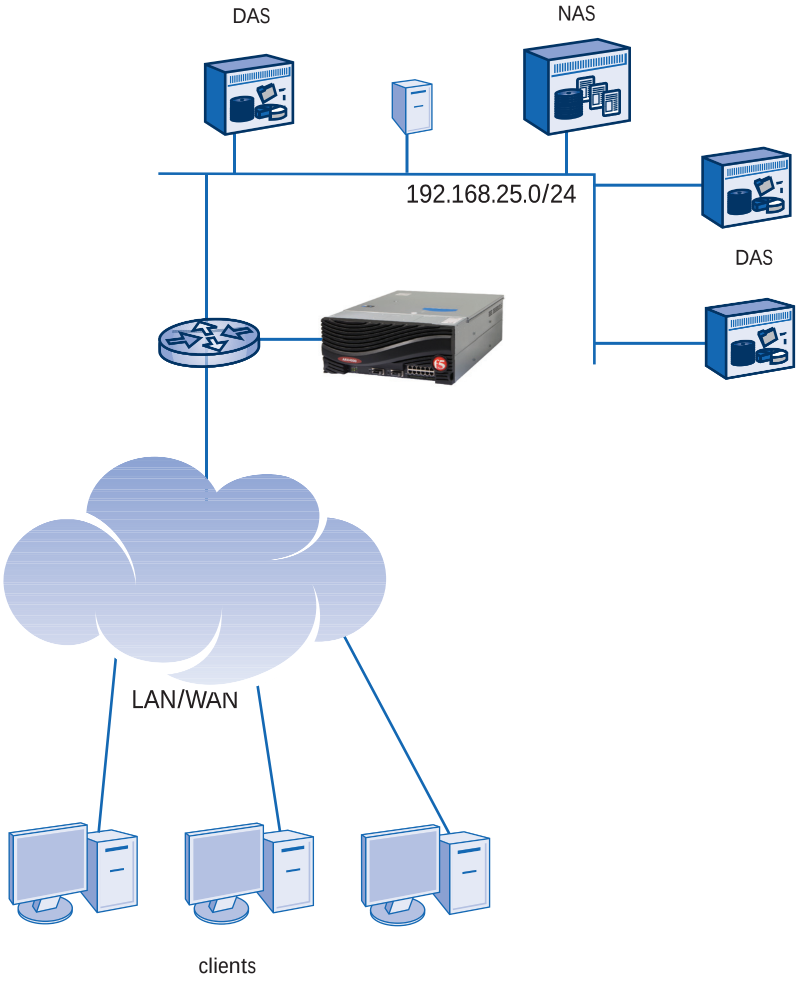
To enter gbl mode, use the global command instead:

```
SWITCH> enable
SWITCH# global
SWITCH(gbl)#
```

The command sequences in this manual all begin either in cfg mode or gbl mode.

Sample Network

The examples in this manual draw from a single, fictitious network. The network filers all live on a class-C subnet at 192.168.25.x. These filers are called *back-end* filers, since they are the storage behind the *front-end* services of the ARX. The filers can be heterogeneous: NAS devices and file servers (possibly with additional DAS) need only support CIFS or NFS to be on the back end of the ARX.



Contacting Customer Service

You can use the following methods to contact F5 Networks Customer Service:

F5 Networks Online Knowledge Base Online repository of answers to frequently-asked questions.	http://support.f5.com
F5 Networks Services Support Online Online customer support request system	https://websupport.f5.com
Telephone	Follow this link for a list of Support numbers: http://www.f5.com/support/support-services/contact/

2

Product Overview

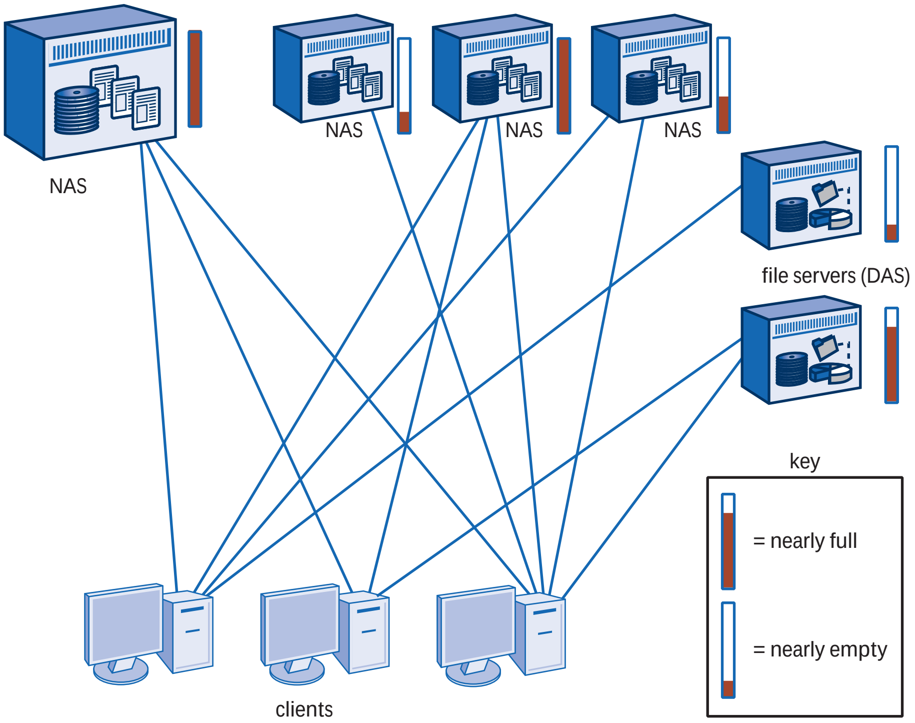
Overview

This chapter shows some of the problems inherent with today's file-storage networks, then it demonstrates the solutions offered by the ARX. References to relevant chapters appear at the end of each solution, so that you can configure the solutions in your network.

Today's File Storage

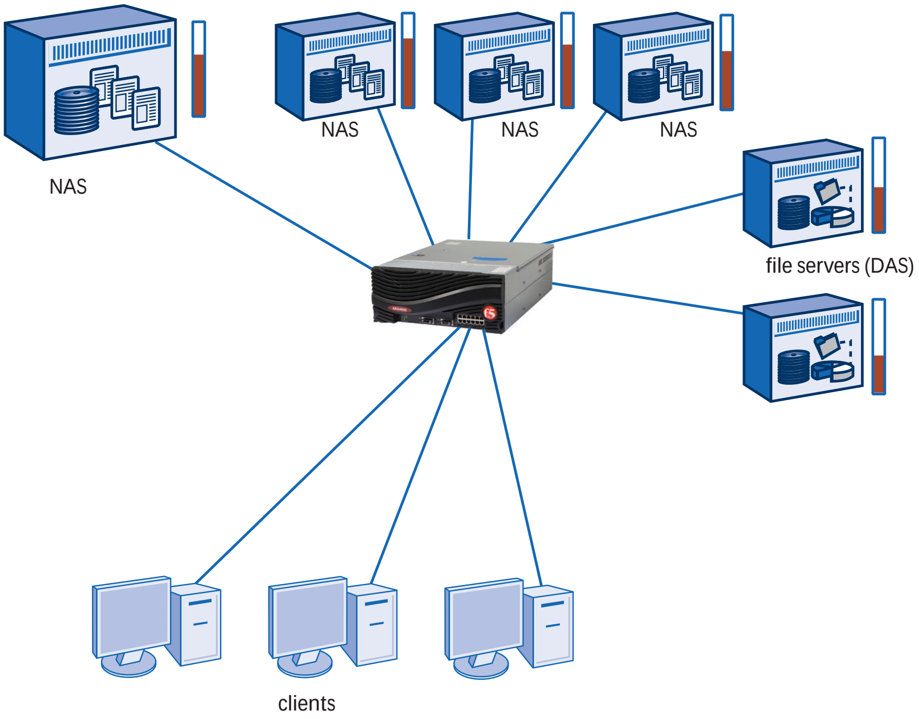
Today's storage networks typically evolve over time into storage *islands* with imbalanced capacities. As you add new storage devices to the network, clients view each device as a discrete set of CIFS shares or NFS exports. Inevitably, some shares are more popular than others, causing load imbalance amongst the storage devices. Storage engineers try to match popular shares with storage devices that can handle the most load, but client preferences change and client writes are unpredictable.

Balancing the capacity between these islands means moving popular files between file servers, but this can be difficult. Each client connects to back-end storage statically, through an IP address or FQDN, and chooses from a list of shares and paths that reside at that storage device. Moving files from one storage device to another means updating the client-side view of IP addresses, share names, and/or file paths. File storage is therefore static and expensive to manage.



Adaptive Resource Switching solves these expensive problems by creating a virtual view of back-end storage for the front-end clients. Clients connect to all back-end storage through a virtual directory structure called a *namespace volume*. The storage devices are hidden behind the volume, making it possible to balance capacity and load at the back end without affecting any client-access on the front-end. An Adaptive Resource Switch can thereby

- *optimize* the file-storage infrastructure,
- *adapt* file storage to client demands, and
- *control* management costs.

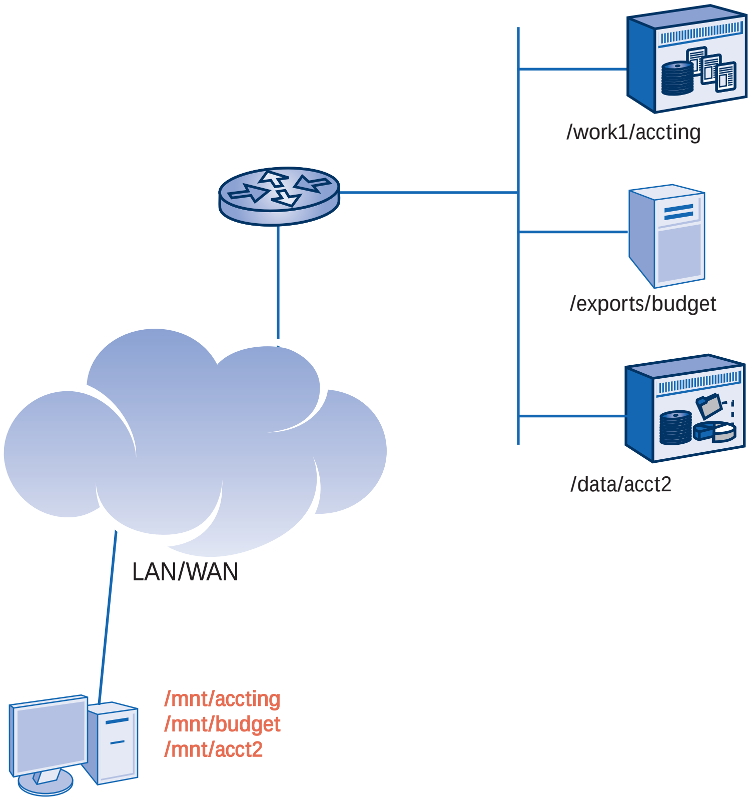


F5's Adaptive Resource Switch can optimize, adapt, and control your storage resources through namespace configuration and file migration. The sections below summarize the configuration steps for each of these solutions.

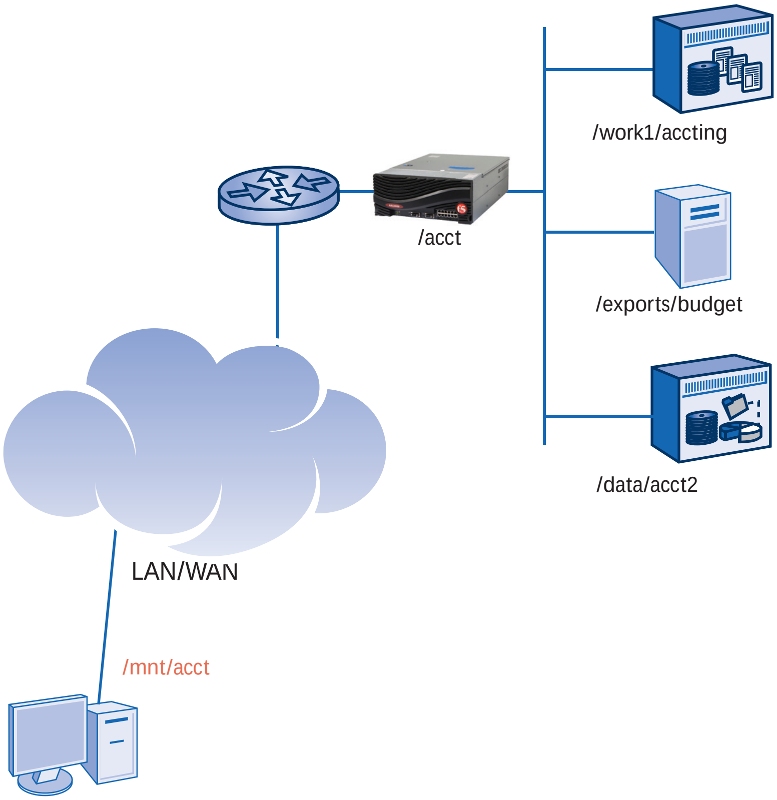
Optimizing Storage in a Namespace Volume

A *namespace* is a group of file systems under a single authentication domain. A namespace is comprised of one or more *volumes*, where each volume is like a discrete file system. A volume is composed of *shares*, where each share maps to an export or share on an actual back-end filer. The volume can contain shares from multiple back-end filers, but the client sees a single mount-point or share-point.

Consider three filers with one NFS export each. The figure below shows the filers behind a standard router. In this configuration, the client must issue three mounts to access all three exports.



The ARX can aggregate all three exports into a single namespace volume, “/acct” in this example. The client then only needs to mount the single, aggregated volume.



The client now connects to the ARX rather than the individual filers. This creates an opportunity for upgrading storage on the back-end without changing the front-end view.

Through storage aggregation in a namespace volume, the ARX simplifies the client’s interface to multiple storage devices and creates new opportunities for storage maintenance.

Configuration Instructions

Namespace configuration constitutes the bulk of switch configuration. You begin by connecting to back-end filers and aggregating their storage.

1. *Chapter 6, Adding an External Filer*, contains instructions for adding an *external* (NAS or DAS-enhanced) filer to the configuration.
2. *Chapter 7, Configuring a Namespace*, contains instructions for aggregating external-filer storage into a namespace.

Once the namespace is configured, you must configure the server for clients to access the namespace:

1. configure a *global server* that clients can use to access the namespace over IP (see *Chapter 10, Configuring a Global Server*), and
2. create one or more front-end services, such as NFS or CIFS, to offer the namespace storage through the global server. See *Chapter 11, Configuring Front-End Services*.

Adapting Storage to User Demands

The ARX's position between clients and back-end servers presents a unique opportunity for adapting to changes in usage. The ARX processes every client request and server response, so it can

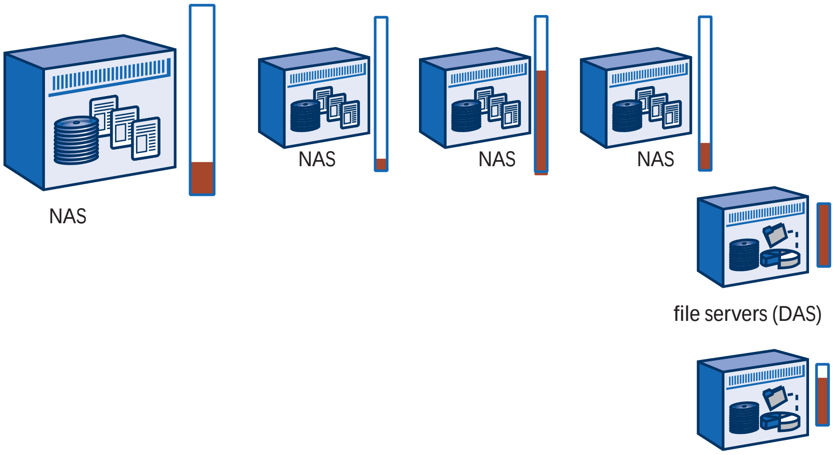
- *classify* information as it is generated,
- *monitor* client-access patterns and back-end-resource usage, and
- *adapt* the back-end resources to meet client demands in real time.

The ARX adapts by migrating files from one filer to another. You use namespace policy to set rules and thresholds for migration.

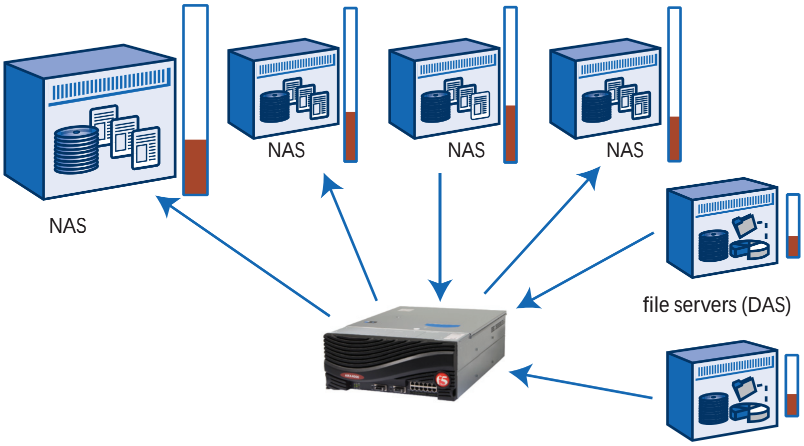
Migration for Capacity

The ARX can keep all filers at or above the same minimum free space, so that overburdened filers can offload their files to other filers. This is called *auto-migration* off of the filer that is low on free space. You configure this by declaring a share farm and establishing the auto migration rule in the share farm. The internal policy engine balances capacity by migrating popular files from an over-filled share to the shares with more free space.

For example, consider a scenario where several larger filers are under-utilized and two DAS-enhanced file servers are nearly filled to capacity:



An auto-migrate rule migrates files off of the over-burdened filers and onto the filers with more available free space. In addition, the ARX ensures that no filer is over-filled; all filers in the share farm maintain the minimum free space until/unless they all fill up to this level.



Configuration Instructions

An auto-migrate rule is one of the share-farm rules described in *Chapter 15, Grouping Shares in a Share Farm*. See *Auto Migrating Existing Files*, on page 15-15.

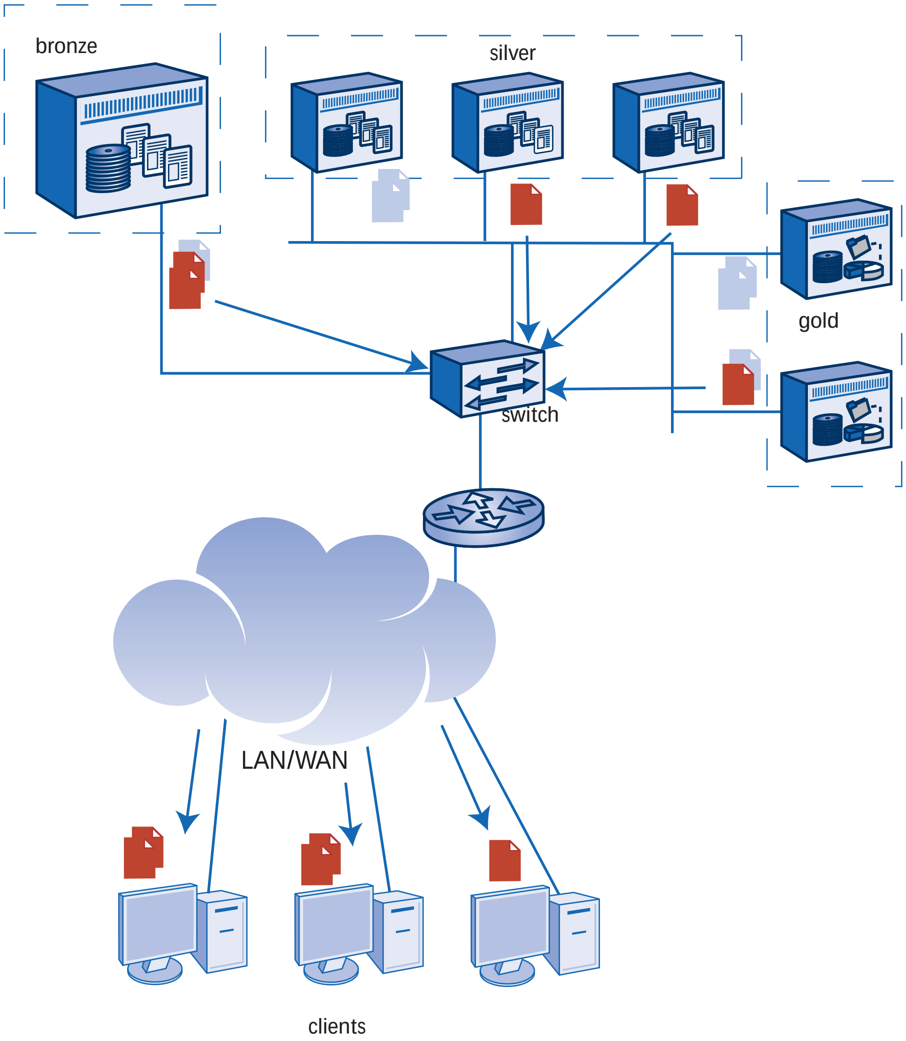
Migration for Class of Storage: File-Placement Policy

Another application for namespace policy is adaptive migration of files onto selected back-end storage. This migration is configured as a *file-placement* policy. To configure file-placement policy, you can

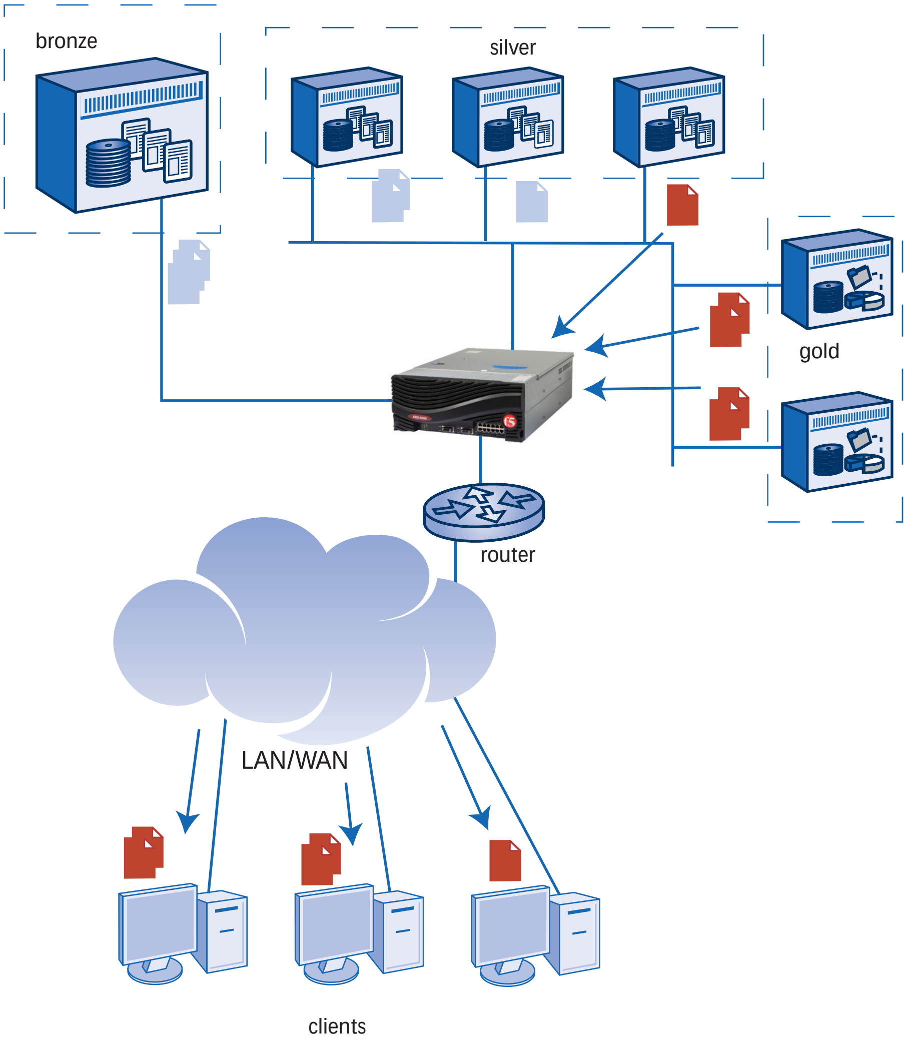
1. group files into *filesets* based on criteria like names (such as *.xml or *.mpg), location (/usr/local), and/or last-modified times,
2. differentiate your back-end filers by classifying them into different *share farms* (optional), and
3. configure a file-placement policy to place a fileset onto a given share or share farm.

The policy engine periodically re-examines its back-end files to re-group them into filesets, then it moves them to the configured back-end share or share farm.

Consider a site with several tiers of storage: a gold tier of expensive file servers, a silver tier of more-plentiful (perhaps slower) filers, and a bronze tier of least-expensive filers. Initially, administrators distribute files among their filers based on best guesses at the usage of the various files. Periodically, files in the bronze tier become unexpectedly “hot,” impacting performance on that tier of the storage network.



File-placement policy can solve this problem. You can configure an age-based fileset that groups all files in the namespace based on their last-accessed times. This fileset could divide the files into weekly groups: files accessed this week, two-to-four weeks ago, and any time before four weeks ago. A file placement policy could then place the files on the correct tiers based on when they were last accessed. The ARX dynamically ensures that the most-popular files reside on the storage best-equipped to serve them.



Configuration Instructions

To configure a fileset, see *Chapter 13, Grouping Files Into Filesets*.

For instructions on moving the fileset to your chosen storage, see *Chapter 14, Migrating Filesets and Tiering*.

To group your namespace shares into a share farm, see *Chapter 15, Grouping Shares in a Share Farm*.

Controlling Costs

The ARX controls costs by optimizing network resources behind a namespace and adapting to client usage in real time. By configuring a namespace and namespace policies, you can

- reclaim existing storage resources (with share-farm balancing),
- expand your storage-purchasing options (by grouping filers into share farms), and
- simplify file-storage management (by hiding filers behind a namespace volume).

3

Preparing for CIFS Authentication

Overview

The ARX is a file proxy between clients and back-end filers; it must authenticate clients on the front end, and it must provide valid credentials to servers on the back end. To set up the switch proxy in an CIFS environment, you must configure at least two authentication parameters in advance:

- A *proxy user*, configured with a valid username and password.

  The switch uses the proxy-user credentials when it performs autonomous operations (such as moving files between shares).
- The structure of the *Active-Directory forest* in your network. This identifies the domain controllers (DCs) for each Windows domain, and it provides the ARX with the Windows-domain hierarchy.

You must configure these Windows-security parameters first so that you can reference them later.

You can also define the permissions for a group of CIFS clients to use Windows-Management tools, such as the Microsoft Management Console (MMC) and snapshot access, in CIFS namespaces. You can create one or many such Windows-management-authorization (*WMA*) groups. You can later assign each group to one or more CIFS namespaces.

Concepts and Terminology

A *volume* is an aggregated view of several back-end filers. A *managed volume* contains metadata that tracks the locations of all back-end files and directories; it uses this metadata to support migration and replication policies. A *direct volume* is a series of mount points to various back-end filers, with no metadata. Direct volumes are less complex than managed volumes, but do not support policies.

A *namespace* is container for one or more volumes. Each namespace operates under a single authentication domain, the same domain supported by all of its back-end filers. This applies to both Windows and Unix domains.

A *global server* is an client-entry point to the services of the ARX. A global server has a fully-qualified domain name (such as myserver.mycompany.com) where clients can access namespace storage.

Adding a Proxy User

Before you configure CIFS authentication for a namespace, you must configure a proxy user for the namespace. A proxy user is a single username/password in a particular Windows domain. The ARX uses the proxy user as its identity while reading from CIFS shares (to import the share into a namespace) and moving files between shares (for capacity balancing and other policies).

◆ **Note**

A Windows proxy user must belong to the Backup Operators group on every filer/server, to ensure that it has sufficient authority to move files freely from share to share. The user must also have "full control" (both read and change privileges) for every file and directory in the back-end share.

If the ARX must also read directory paths for a share, the proxy user must belong to the filer's more-privileged Administrators group. Directory-path knowledge is required to support CIFS subshares, described later in this manual.

For back-end filers that support User Access Control (UAC), the proxy user must belong to the domain-level Users group.

From gbl mode, use the proxy-user command to create one proxy user:

```
proxy-user name
```

where ***name*** (1-32 characters) is a name you choose.

This puts you into gbl-proxy-user mode, where you set the Windows domain, username, and password for the proxy user. When you later configure a namespace in the same Windows domain, you can apply this proxy-user configuration to that namespace. You can apply the same proxy user to multiple namespaces in the same domain.

For example, the following command sequence creates a proxy user named "acoProxy2:"

```
bstnA(gbl)# proxy-user acoProxy2
bstnA(gbl-proxy-user[acoProxy2])# ...
```

Specifying the Windows Domain

The first step in configuring a proxy user is to specify its Windows domain. From gbl-proxy-user mode, use the windows-domain command to specify the domain:

```
windows-domain name
```

where ***name*** is 1-64 characters. Enter a complete FQDN (such as "myco.com") for the ARX to use Kerberos for its proxy-user authentications. The ARX uses the first part of the FQDN (such as "myco", above) if it requires a NetBIOS-style name.

For example, the following command sequence specifies the “MEDARCH.ORG” domain for the “acoProxy2” proxy user and another domain for the “acoProxy3” proxy user:

```
bstnA(gbl)# proxy-user acoProxy2
bstnA(gbl-proxy-user[acoProxy2])# windows-domain MEDARCH.ORG
bstnA(gbl-proxy-user[acoProxy2])# exit
bstnA(gbl)# proxy-user acoProxy3
bstnA(gbl-proxy-user[acoProxy3])# windows-domain FDTESTNET.COM
bstnA(gbl-proxy-user[acoProxy3])# ...
```

Using a Pre-Windows-2000 Domain Name

Some back-end filers cannot accept an FQDN (such as “FDTESTNET.COM” in the example above) for the ARX’s NTLM authentications. These filers accept only a domain name format used prior to the release of Windows 2000: 1-15 bytes and no periods (“.”). All filers accept this old-style domain format, known to the ARX as “pre-win2k,” so the ARX performs its NTLM authentications with the first part of the FQDN you typed above. This is the name before the first period (“.”) in the FQDN, up to 15 characters. In the example above, this would be “FDTESTNET.”

The pre-win2k name is used when the environment does not use Active Directory, or does not support Kerberos authentication. For example, the client may be a computer using a pre-win2k version of Windows, or some clients that are joined to the domain may not have a forest trust with the same forest to which the ARX CIFS service and filers are joined.

If the pre-win2k domain name is not configured explicitly, the pre-win2k domain name is discovered automatically during AD configuration and/or AD discovery. In the unlikely event that a pre-win2k domain name is not identified at that time, the ARX will derive a pre-win2k domain name as a last resort. This is done by truncating the FQDN to the first 15 characters before the first period.

For a domain that uses a different pre-Windows-2000 name, you can use the optional pre-win2k-name argument to configure a pre-win2k domain name explicitly, but this should not be necessary if Active Directory is in use:

```
windows-domain name pre-win2k-name old-style-name
```

where ***old-style-name*** is 1-15 characters and does not contain any periods.

For example, the following command sequence enters “BOSTONCIFS,” a pre-Windows-2000 domain name, for the “acoProxy3” proxy user:

```
bstnA(gbl)# proxy-user acoProxy3
bstnA(gbl-proxy-user[acoProxy3])# windows-domain FDTESTNET.COM pre-win2k-name BOSTONCIFS
bstnA(gbl-proxy-user[acoProxy3])# ...
```

The ARX will use available pre-win2k domain names in the following precedence order:

1. A pre-win2k domain name configured explicitly using the pre-win2k-name option. This should not be necessary.

2. A pre-win2k domain name discovered automatically during AD configuration and/or AD discovery. This is the default behavior, and should work reliably for most implementations.
3. A pre-win2k domain name derived by truncating the FQDN to the first 15 characters before the first period. The ARX does this as a last resort.

Specifying the Username and Password

The final step in configuring a proxy user is to specify a username and password. The username must be for a user in the above Windows domain. On all of your CIFS filers, this user must belong to the Backup Operators group to ensure that it has sufficient authority to move files freely from share to share. If the proxy user will have to read directory paths on the filer (to support CIFS subshares or Access-Based Enumeration, discussed later in the manual), the proxy user should belong to the more-privileged Administrators group. If you use *local* groups in a Windows cluster, add the user to the local group at every node in the cluster.

Use a domain-level user account (not a local one) if any of your Windows filers support User Account Control (UAC). UAC is a security feature introduced with Windows Server 2008.

From gbl-proxy-user mode, use the user command to specify the username:

```
user username
```

where ***username*** (1-64 characters) is a valid username in the proxy-user’s domain.

The CLI prompts you for the user’s password, then prompts to validate the password.

For example, the following command sequence specifies the username, “jqpublic:”

```
bstnA(gbl-proxy-user[acoProxy2])# user jqpublic
Password: jqpasswd
Validate Password: jqpasswd
bstnA(gbl-proxy-user[acoProxy2])# ...
```

Adding a Description

You can add a description to the proxy-user configuration, for use in the show command below. The description can differentiate the proxy user from others. From gbl-proxy-user mode, use the description command to describe the proxy user:

```
description text
```

where ***text*** is 1-255 characters. Quote the text if it contains any spaces.

For example:

```
bstnA(gbl)# proxy-user acoProxy2
bstnA(gbl-proxy-user[acoProxy2])# description "user with backup and admin creds on our servers"
bstnA(gbl-proxy-user[acoProxy2])# ...
```

Removing the Description

From gbl-proxy-user mode, use no description to remove the description string:

no description

For example:

```
bstnA(gbl)# proxy-user acoProxy3
bstnA(gbl-proxy-user[acoProxy3])# no description
bstnA(gbl-proxy-user[acoProxy3])# ...
```

Listing All Proxy Users

You can use the show proxy-user command to get a list of all proxy users on the ARX:

show proxy-user

For example:

```
bstnA(gbl)# show proxy-user

Name             Windows Domain                Pre-Win2k        User Name
                 Description
--------------------------------------------------------------------------------
acoProxy1        WWMEDNET.COM                  WWMEDNET         jqprivate
                 jq's admin account

acoProxy3        FDTESTNET.COM                 BOSTONCIFS       jqtester

cifs_admin       MEDARCH.ORG                   MEDARCH          Administrator

nas_admin                                                       root

ny_admin         NY.COM                        NY               jqpublic

acoProxy2        MEDARCH.ORG                   MEDARCH          jqpublic
                 user with backup and admin creds on our servers

bstnA(gbl)# ...
```

Showing One Proxy User

To focus on one proxy user, you can specify a name in the show proxy-user command:

show proxy-user *name*

where ***name*** (1-32 characters) identifies the proxy user to show.

For example:

```
bstnA(gbl)# show proxy-user acoProxy2

Name             Windows Domain                Pre-Win2k        User Name
                 Description
--------------------------------------------------------------------------------
acoProxy2        MEDARCH.ORG                   MEDARCH          jqpublic
```

```
                user with backup and admin creds on our servers

bstnA(gbl)# ...
```

Removing a Proxy User

From gbl mode, use no proxy-user to remove a proxy-user configuration:

no proxy-user ***name***

where ***name*** (1-32 characters) identifies the proxy user to remove.

For example, the following command sequence removes a proxy user called proxyNYC:

```
bstnA(gbl)# no proxy-user proxyNYC
```

Discovering the Active-Directory Forest (Kerberos)

To prepare for a CIFS service that uses Kerberos authentication, you must first identify the *Active Directory (AD) forest* in your Windows network. When a client accesses the CIFS front-end service from one of the domains in the AD forest, the switch uses this information to locate the appropriate DC for authentication.

You can skip this section if you are not using Kerberos with any CIFS service.

Given one of the domains in the AD forest, called the *seed domain*, the ARX can automatically discover all of the domains and DCs in the forest. The discovery operation performs a DNS lookup to find the seed domain's DC, then queries the DC for the names of other DCs in the same forest. The DC uses FQDNs instead of IP addresses; the ARX must perform more DNS lookups to translate those names into IP addresses.

Before you start discovering AD forests, configure the switch to perform DNS lookups at a properly-configured DNS server. Refer to the *ARX® CLI Network-Management Guide*: see *Configuring DNS Lookups*, on page 4-31. Choose a DNS server that can perform lookups for all of the domains in the desired forest.

Once you have DNS configured, you can use the active-directory update command to discover the AD forest automatically. This command is in priv-exec mode:

```
active-directory update seed-domain seed proxy-user proxy
[domain-controllers max-dcs] [site-name site] [verbose] [tentative]
```

where

seed (1-255 characters) is the name of one domain in the forest. The ARX uses this domain name to begin its forest discovery. This becomes the name of the AD-forest in the ARX configuration.

proxy (1-32 characters) is a proxy user with credentials for accessing the seed domain's DC(s). These credentials can belong to the seed domain itself, or any domain that is trusted by the seed domain. The ARX queries the DC for the names of other domains in the same AD forest.

max-dcs (optional, 1-100) sets a maximum number of DCs used in each domain. The ARX queries its DNS server to discover all the DCs in each domain; if the DNS server returns more DCs than ***max-dcs***, the ARX takes the top DCs from the DNS list. The ARX uses the order returned from DNS.

site (optional, 1-64 characters) Use this option if the AD's site configuration does not include the proxy-IP subnet. This identifies the AD site for the ARX. If the ARX knows of multiple DCs that can answer the same query, it prefers DCs in its own site (if there are any) over DCs in any other site. The site name is defined on a DC with the Active Directory Sites and Services plugin. The site name is case insensitive, so “boston” and “BOSTON” are equivalent. If you omit this option, the ARX software uses the AD site configured for the *ip proxy-address* subnet.

verbose (optional) causes the command to show the results of the forest discovery as it progresses.

tentative (optional) makes the ARX perform the AD-forest discovery without creating the actual *active-directory-forest* configuration.

This command generates a report with details about the discovery process. This includes all of the information that you see when you use the **verbose** flag. The CLI displays the name of the report after you issue the command. Use show reports *report-name* to read the report.

For example, the following command automatically discovers the “MEDARCH.ORG” forest:

```
bstnA# active-directory update seed-domain medarch.org proxy-user acoProxy2

Report File : active-directory-MEDARCH.ORG.rpt
bstnA# . . .
```

Editing the AD-Forest Configuration

Some components of the AD-forest configuration cannot be discovered dynamically. These are described in the sections below. You edit the configuration through gbl-forest mode, which you can enter with the active-directory-forest command:

```
active-directory-forest forest-name
```

where ***forest-name*** (1-256 characters) is the name of the forest. This is the seed domain that you used to discover the forest, above.

For example, the following command edits the ‘medarch.org’ forest:

```
bstnA(gbl)# active-directory-forest medarch.org
bstnA(gbl-forest[medarch.org])# . . .
```

Identifying a Dynamic-DNS Server

Many Active-Directory networks use dynamic DNS to map CIFS host names to IP addresses. Whenever an ARX’s front-end CIFS service changes its host name or IP address, the CIFS service can use dynamic DNS to inform the servers in the AD forest. No manual changes to the DNS configuration are required.

Up-to-date DNS configuration is required for Kerberos, which uses FQDNs in its authentication tickets instead of IP addresses.

RFCs 1034 and 1035 define basic DNS, and RFC 3645 defines the Microsoft-specific authentication extensions for dynamic DNS. The ARX implementation supports all of these standards; it does not support any other dynamic-DNS RFCs.

To prepare for dynamic DNS, you identify the dynamic-DNS servers in this forest. (The active-directory update command cannot automatically discover dynamic-DNS servers.) Later chapters explain how to configure a front-end CIFS service to use these dynamic-DNS servers. To identify one dynamic-DNS server, called a *name server*, use the name-server command in gbl-forest mode:

```
name-server domain-name ip-address
```

where

domain-name (1-255 characters) identifies the AD domain, and

ip-address is the IP address of the domain's name server. This might be the same IP as for the domain's DC; dynamic DNS often runs on the same DC that supports the domain.

You can enter this command multiple times, once for each name server. If you enter multiple name servers for a given AD domain, a CIFS service in this domain will attempt to register with each of them in turn until it succeeds. It stops registering on the first success.

For example, this command sequence identifies three dynamic-DNS servers for the 'MEDARCH.ORG' domain. The first, 192.168.25.102, is also the DC for the forest root:

```
bstnA(gbl)# active-directory-forest medarch.org
bstnA(gbl-forest[medarch.org])# name-server MEDARCH.ORG 192.168.25.102
bstnA(gbl-forest[medarch.org])# name-server MEDARCH.ORG 192.168.25.103
bstnA(gbl-forest[medarch.org])# name-server MEDARCH.ORG 192.168.25.104
bstnA(gbl-forest[medarch.org])# . . .
```

Removing a Name Server

If you remove the only name server for an Active-Directory domain, any changes to front-end CIFS services in that domain will require manual updates to DNS. That is, if a CIFS service is added to or removed from the ARX, an administrator must add or remove the corresponding "A" record from the external DNS server. As long as at least one name server is assigned to the domain, this maintenance penalty is not necessary. The DNS database must be correct for the CIFS service or Windows clients cannot authenticate with Kerberos.

To remove a name server from an AD domain, use the no name-server command:

```
no name-server domain-name domain-controller
```

where

domain-name (1-255 characters) identifies the AD domain, and

domain-controller is the IP address of the name server to remove.

For example, this command sequence removes the second (redundant) name server from the 'MEDARCH.ORG' domain. Recall from the previous example that this leaves two more name servers for the domain.

```
bstnA(gbl)# active-directory-forest medarch.org
bstnA(gbl-forest[medarch.org])# no name-server MEDARCH.ORG 192.168.25.103
bstnA(gbl-forest[medarch.org])# . . .
```

Updating the ARX View of the AD Forest

You can run a different form of the active-directory update command to update the existing AD-forest configuration on the ARX. This update operation changes the ARX configuration so that it conforms to the current state of the AD forest; it adds, changes, or removes DCs depending on what it finds in the actual network.

◆ **Note**

The active-directory-update operation overwrites the previous AD-forest configuration on the ARX, whether or not you set it manually. This could possibly be an issue if you set a ***max-dcs*** *value in the command; the ARX discards any manually-configured DCs and replaces them with the first n DCs returned from DNS.*

To update the ARX view of an AD forest, use the active-directory update command with the forest option (instead of the seed-domain option described earlier):

active-directory update forest *forest-name* proxy-user *proxy* [domain-controllers *max-dcs*] [site-name *site*] [verbose] [tentative]

where

forest-name (1-255 characters) is the name of an already-configured AD forest, and

the remaining options were described previously.

This generates a similar report to the one created with the initial seed-domain command. The report has the same name, "active-directory-*forest-name*.rpt," where *forest-name* is the same as the one used in the command.

For example, the following command automatically updates the ARX view of the "MEDARCH.ORG" forest:

```
bstnA(gbl)# end
bstnA# active-directory update forest medarch.org proxy-user acoProxy2

Report File : active-directory-MEDARCH.ORG.rpt
bstnA# . . .
```

Scheduling Regular Updates

If your site makes smaller, incremental changes to the AD forest, you can use the at command to arrange for regular AD-forest updates. The at command is in cfg mode.

For example, the following command sequence runs the active-directory update command every morning at 3 AM:

```
bstnA(gbl)# end
bstnA# config
bstnA(cfg)# at 03:00:00 every 1 days do "active-directory update forest NY.COM proxy-user
ny_admin"

  The scheduled execution time for CLI command is: 8/30/07 03:00 AM.
```

```
bstnA(cfg)# ...
```

Reviewing the Report

Each AD update creates one report to show its results. The ARX names the AD-discovery reports according to the following convention:

active-directory-*domain*.rpt

where *domain* is the name of the root domain in the AD forest.

To conserve disk space, each AD update overwrites any previous reports.

Use the show reports type adUp command to list all AD-update reports.

`show reports type AdUp`

Use show reports *report-name*, tail, or grep to read the file. To save the report off to an external FTP site, use the copy ... ftp command from priv-exec mode. To upload the report to an SCP host, use copy ... scp. All of these commands are documented in the *CLI Reference* manual.

This report is divided into three sections: a discovery section, a list of resultant changes in the ARX view of the AD forest, and the final AD-forest configuration on the ARX. The discovery process finds all domains in the forest and then categorizes the domains into various “types,” described later.

For example, this shows an AD-update report for “MEDARCH.ORG:”

```
bstnA(gbl)# show reports active-directory-MEDARCH.ORG.rpt
**** Active-Directory Update Report: Started at 04/11/2012 00:43:52 -0400 ****
**** Software Version: 6.02.000.14353 (Apr  6 2012 20:12:43) [nbuilds]
**** Hardware Platform: ARX-4000
**** Report Destination:

**** Command : active-directory update seed-domain MEDARCH.ORG            proxy-user acoProxy2

DNS lookup MEDARCH.ORG            from 192.168.25.102
    192.168.25.109
    192.168.25.104
    192.168.25.110
    192.168.25.102
    192.168.25.111

Enumerate Forest MEDARCH.ORG            from 192.168.25.109
    Domain Name                                  Pre-Win2k Name       Domain Type
    -------------------------------------------  --------------------  -------------
    bostoncifs.fdtestnet.net                     BOSTONCIFS            child-domain
    ma.ne.medarch.org                            MA                    child-domain
    westcoast.medarch.org                        WESTCOAST             child-domain
    ne.medarch.org                               NE                    child-domain
    MEDARCH.ORG                                  MEDARCH               tree-domain
    bostonmed.org                                BOSTONMED             tree-domain
    fdtestnet.net                                FDTESTNET             tree-domain

Current Configuration:


  active-directory-forest MEDARCH.ORG
```

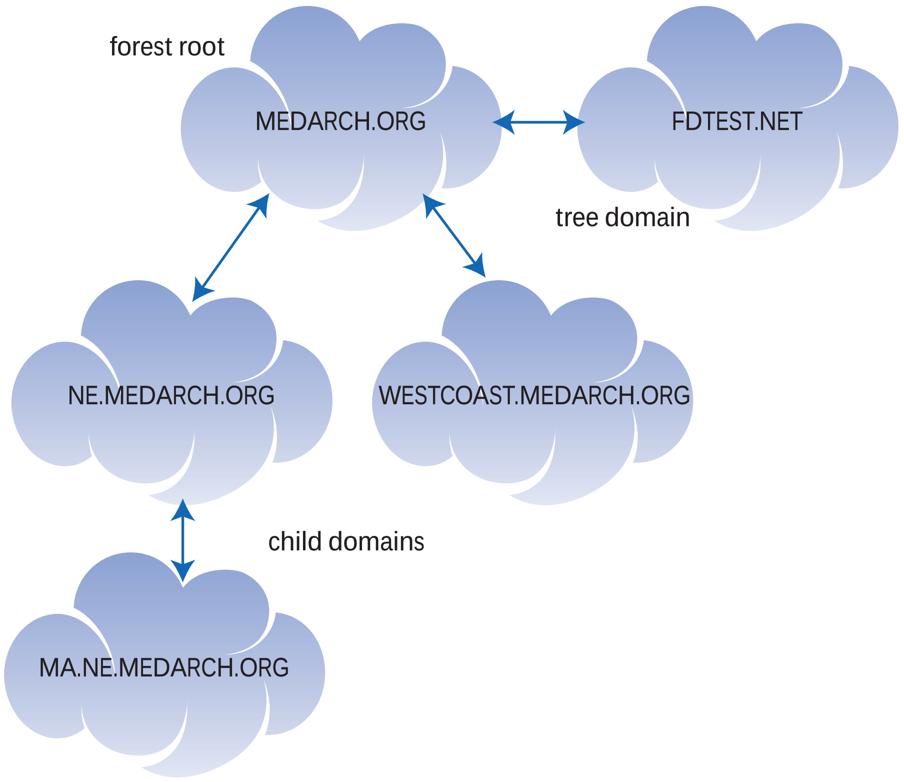
```
    exit


Change Summary:

    Site:  Default-First-Site-Name


    tree-domain fdtestnet.net FDTESTNET 172.16.168.22added as preferred.
    tree-domain fdtestnet.net FDTESTNET 172.16.168.21 added as preferred.
    tree-domain bostonmed.org      BOSTONMED 172.16.74.88 added as preferred.
    tree-domain bostonmed.org      BOSTONMED 172.16.74.89 added as preferred.
    forest-root MEDARCH.ORG            MEDARCH   192.168.25.102 added as preferred.
    forest-root MEDARCH.ORG            MEDARCH   192.168.25.111 added as preferred.
    forest-root MEDARCH.ORG            MEDARCH   192.168.25.109 added as preferred.
    forest-root MEDARCH.ORG            MEDARCH   192.168.25.110 added as preferred.
    forest-root MEDARCH.ORG            MEDARCH   192.168.25.104 added as preferred.
    child-domain ne.medarch.org               NE        172.16.124.19added as preferred.
    child-domain ne.medarch.org               NE        172.16.124.73 added as preferred.
    child-domain westcoast.medarch.org       WESTCOAST 192.168.202.11 added as preferred.
    child-domain westcoast.medarch.org       WESTCOAST 192.168.202.9 added as preferred.
    child-domain westcoast.medarch.org       WESTCOAST 192.168.202.10 added as preferred.
    child-domain westcoast.medarch.org       WESTCOAST 192.168.202.16 added as preferred.
    child-domain ma.ne.medarch.org                   MA        192.168.25.105 added as preferred.
    child-domain ma.ne.medarch.org                   MA        192.168.25.103 added as preferred.
    child-domain bostoncifs.fdtestnet.net BOSTONCIFS10.19.230.74 added as preferred.
    child-domain bostoncifs.fdtestnet.net BOSTONCIFS10.19.230.94 added as preferred.
    child-domain bostoncifs.fdtestnet.net BOSTONCIFS10.19.230.88 added as preferred.

New Installed Configuration:


  active-directory-forest MEDARCH.ORG
    forest-root    MEDARCH.ORG                             MEDARCH           192.168.25.102
preferred
    forest-root    MEDARCH.ORG                             MEDARCH           192.168.25.111
preferred
    forest-root    MEDARCH.ORG                             MEDARCH           192.168.25.109
preferred
    forest-root    MEDARCH.ORG                             MEDARCH           192.168.25.110
preferred
    forest-root    MEDARCH.ORG                             MEDARCH           192.168.25.104
preferred
    tree-domain    fdtestnet.net                           FDTESTNET         172.16.168.22
preferred
    tree-domain    fdtestnet.net                           FDTESTNET         172.16.168.21
preferred
    tree-domain    bostonmed.org                           BOSTONMED         172.16.74.88
preferred
    tree-domain    bostonmed.org                           BOSTONMED         172.16.74.89
preferred
    child-domain   bostoncifs.fdtestnet.net                BOSTONCIFS        10.19.230.74
preferred
    child-domain   bostoncifs.fdtestnet.net                BOSTONCIFS        10.19.230.94
preferred
    child-domain   bostoncifs.fdtestnet.net                BOSTONCIFS        10.19.230.88
preferred
    child-domain   ne.medarch.org                          NE                172.16.124.19
preferred
    child-domain   ne.medarch.org                          NE                172.16.124.73
preferred
```

```
    child-domain   westcoast.medarch.org                   WESTCOAST         192.168.202.11
preferred
    child-domain   westcoast.medarch.org                   WESTCOAST         192.168.202.9
preferred
    child-domain   westcoast.medarch.org                   WESTCOAST         192.168.202.10
preferred
    child-domain   westcoast.medarch.org                   WESTCOAST         192.168.202.16
preferred
    child-domain   ma.ne.medarch.org                       MA                192.168.25.105
preferred
    child-domain   ma.ne.medarch.org                       MA                192.168.25.103
preferred

    exit

**** Total processed:              20
**** Elapsed time:           00:00:00
**** AD Update Report: DONE at 04/11/2012 00:43:52 -0400 ****

bstnA(gbl)# ...
```

Domain Types in an AD Forest

Each domain in the AD forest has one of three *types*: the forests *root domain*, a *child domain*, or a *tree domain*. The root domain is the root of the main “tree” in the forest (for example, “myco.com”), and a child domain is the child of the root or some other child domain. Parent-to-child relationships are determined by the domains’ names; “mydebt.myco.com” is a child of “myco.com.” A tree domain is the root of another domain tree in the same forest. The tree-domain name seldom has any relationship to the name of the root domain: for example, “yourorg.org” could be a tree domain in the same AD forest with “myco.com.” A tree domain can also have child domains, such as “yourgroup.yourorg.org.”

This illustration shows the root domain, one tree domain, and three child domains in the "MEDARCH.ORG" forest discovered in earlier examples:



Establishing Forest-to-Forest Trust

You can configure multiple AD forests, and, if they have Windows 2003 DCs, establish trust relationships between them. A client from one forest can access a CIFS service in another forest if the forests have a trust relationship. Windows began supporting forest-to-forest trusts in Windows 2003, so this feature is only supported in forests with Windows 2003 DCs.

Forest-to-forest trust must be established at the relevant DCs in order to be effective for the ARX. In addition, you may find it useful to enable selective authentication on the DCs, as a means of limiting the trust relationships between domains in the forests. Selective authentication controls which forest X clients have access privileges in forest Y.

Forest-to-forest trust and selective authentication are enabled on the ARX by default.

Showing All Active-Directory Forests

Use the show active-directory command to show all AD forests and forest trusts on this switch.

show active-directory

This also shows a separate table of forest trusts, if there are any.

For example:

```
bstnA(gbl)# show active-directory
Active Directory Domains
------------------------

Forest Name:  MEDARCH.ORG
  forest-root    MEDARCH.ORG             MEDARCH
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    192.168.25.102        KDC DNS          YES
    192.168.25.104        KDC DNS          YES
    192.168.25.111        KDC              YES
    192.168.25.109        KDC              YES
    192.168.25.110        KDC              YES

  tree-domain    bostonmed.org         BOSTONMED
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.74.89          KDC              YES
    172.16.74.88          KDC              YES

  tree-domain    fdtestnet.net         FDTESTNET
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.168.22         KDC              YES
    172.16.168.21         KDC              YES

  child-domain   boostoncifs.fdtestnet.net       BOSTONCIFS
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    10.19.230.88          KDC              YES
    10.19.230.94          KDC              YES

  child-domain   westcoast.medarch.org        WESTCOAST
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    192.168.202.9         KDC              YES
    192.168.202.10        KDC              YES
    192.168.202.11        KDC              YES

  child-domain   ma.ne.medarch.org                    MA
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    192.168.25.105        KDC              YES
    192.168.25.103        KDC              YES

  child-domain   ne.medarch.org                NE
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.124.19         KDC              YES
    172.16.124.73         KDC              YES


Forest Name:  WELLS.ME.ORG
  forest-root    wells.me.org           WELLS
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.108.136        KDC              NO
    172.16.108.139        KDC              NO

  child-domain   adk.wells.me.org      ADK
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.110.8          KDC              NO
    172.16.110.5          KDC              NO

  child-domain   york.wells.me.org     YORK
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.120.22         KDC              NO
    172.16.120.5          KDC              NO


Forest Name:  VT.COM
  forest-root    VT.COM                VT
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.213.8          KDC              NO

  tree-domain    ATLANTIC.ME.ORG       ATLANTIC
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.210.14         KDC              NO
    172.16.210.7          KDC              NO

  child-domain   MCNIELS.VT.COM        MCNIELS
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.240.70         KDC              NO
    172.16.240.88         KDC              NO

  child-domain   BSH.ATLANTIC.ME.ORG   BASSHARBOR
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    10.51.100.6           KDC              NO
    10.52.160.1           KDC              NO

bstnA(gbl)#
```

Showing One Active-Directory Forest

To focus on a single AD forest, use the forest keyword at the end of the show active-directory command.

show active-directory forest *forest-name*

where ***forest-name*** (1-256 characters) identifies the forest to show.

For example:

```
bstnA(gbl)# show active-directory forest MEDARCH.ORG
Active Directory Domains
------------------------

Forest Name:  MEDARCH.ORG
  forest-root    MEDARCH.ORG             MEDARCH
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    192.168.25.102        KDC DNS          YES
    192.168.25.104        KDC DNS          YES
    192.168.25.111        KDC              YES
    192.168.25.109        KDC              YES
    192.168.25.110        KDC              YES

  tree-domain    bostonmed.org         BOSTONMED
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.74.89          KDC              YES
    172.16.74.88          KDC              YES

  tree-domain    fdtestnet.net         FDTESTNET
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.168.22         KDC              YES
    172.16.168.21         KDC              YES

  child-domain   bostoncifs.fdtestnet.net       BOSTONCIFS
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    10.19.230.88          KDC              YES
    10.19.230.94          KDC              YES

  child-domain   westcoast.medarch.org        WESTCOAST
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    192.168.202.9         KDC              YES
    192.168.202.10        KDC              YES
    192.168.202.11        KDC              YES

  child-domain   ma.ne.medarch.org                    MA
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    192.168.25.105        KDC              YES
    192.168.25.103        KDC              YES

  child-domain   ne.medarch.org                NE
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    172.16.124.19         KDC              YES
    172.16.124.73         KDC              YES
```

Showing One Active-Directory Domain

To focus on a single domain, use the domain keyword at the end of the show active-directory command.

show active-directory domain *domain-name*

where ***domain-name*** (1-256 characters) identifies the domain to show.

For example:

```
bstnA(gbl)# show active-directory domain MA.NE.MEDARCH.ORG
Active Directory Domains
------------------------

Forest Name:  MEDARCH.ORG
  child-domain   ma.ne.medarch.org                    MA
    IP Address            Services         Preferred
    --------------------  ---------------  ------------
    192.168.25.105       KDC              YES
    192.168.25.103       KDC              YES
```

Preferring a DC

For each domain, the Kerberos process divides its DCs into two categories: *preferred* and *non-preferred*. The Kerberos process selects its active DC from the preferred list. The active DC processes all Kerberos authentications indefinitely, or until a communication failure occurs.

Every 60 seconds, the Kerberos process sends a simple LDAP query to all DCs in the AD forest. DCs that are too slow to respond are declared "offline," though the Kerberos process continues to send LDAP queries to all of them. If the active DC goes offline, the Kerberos process chooses another DC from the preferred list. The Kerberos process chooses from the non-preferred list if (and only if) there are no preferred DCs available.

By default, no DC is on the preferred list. From gbl-forest mode, you can use one of three commands to place a DC on the preferred list for its domain:

```
{forest-root | tree-domain | child-domain} domain-name domain-controller [preferred]
```

where

forest-root | tree-domain | child-domain is a required choice, based on the domain's placement in the forest hierarchy. You can use the output of show active-directory (described above) to determine the correct domain type.

domain-name (1-255 characters) identifies the domain, and

domain-controller is the IP address of the DC.

preferred (optional) adds the DC to the preferred list. If this is omitted, the DC is demoted to (or remains on) the non-preferred list.

For example, this command sequence adds three DCs to the preferred list for the 'MEDARCH.ORG' domain:

```
bstnA(gbl)# active-directory-forest medarch.org
bstnA(gbl-forest[medarch.org])# forest-root MEDARCH.ORG 192.168.25.102 preferred
bstnA(gbl-forest[medarch.org])# forest-root MEDARCH.ORG 192.168.25.104 preferred
bstnA(gbl-forest[medarch.org])# forest-root MEDARCH.ORG 192.168.25.109 preferred
bstnA(gbl-forest[medarch.org])# . . .
```

As another example, this command sequence puts a single DC on the preferred list for the 'westcoast.medarch.org' domain:

```
bstnA(gbl)# active-directory-forest medarch.org
bstnA(gbl-forest[medarch.org])# child-domain westcoast.medarch.org 192.168.202.9 preferred
bstnA(gbl-forest[medarch.org])# . . .
```

You can run these commands at any time to add a DC to the preferred list.

Removing a DC from the Preferred List

As mentioned above, you can remove the preferred keyword from any of the commands to demote a DC to the non-preferred list:

```
{forest-root | tree-domain | child-domain} domain-name domain-controller
```

where the absence of the preferred flag indicates that the DC is not preferred for active use.

For example, this command sequence demotes the DC at 192.168.25.109:

```
bstnA(gbl)# active-directory-forest medarch.org
bstnA(gbl-forest[medarch.org])# forest-root MEDARCH.ORG 192.168.25.109
bstnA(gbl-forest[medarch.org])# . . .
```

You can run these commands at any time to remove a DC from the preferred list. Use these commands (with and/or without the preferred keyword) to change the currently-active DC in real time.

Load-Balancing Kerberos Requests Across Multiple Domain Controllers

Kerberos authentication requests are distributed (load-balanced) across all of the online DCs in a domain. The set of domain controllers across which the requests are load-balanced is the set of all DCs that are preferred and online, or, if no preferred DCs are online, the set of all DCs that are non-preferred and online.

Two CLI commands assist with monitoring this behavior: show statistics domain-controller load-balancing and clear statistics domain-controller load-balancing.

For example:

```
bstnA# show statistics domain-controller load-balancing

Domain: ADK.WELLS.ME.ORG

DC IP Address    Kerberos Requests   Preferred
---------------  ------------------  ----
172.16.110.8     0                   No
172.16.110.5     0                   No
.
.
.

Domain: MEDARCH.ORG

DC IP Address    Kerberos Requests   Preferred
---------------  ------------------  ----
192.168.25.109   592                 Yes
192.168.25.110   624                 Yes
192.168.25.111   625                 Yes
192.168.25.102   632                 Yes
192.168.25.104   633                 Yes

.
.
.
```

The command clear statistics domain-controller load-balancing causes all of the DC load balancing statistics to be reset to zero:

```
bstnA# clear statistics domain-controller load-balancing
```

Resetting the Threshold for Health Checks

The ARX regularly tests the latency between itself and each of its DCs. It sends a simple LDAP query to each DC every 60 seconds, and it measures the time required for the DC to respond. If any DC takes longer than 2 seconds, the Kerberos software declares it offline and issues an SNMP trap. You can change the 2-second threshold from gbl mode, using the kerberos health-check threshold command:

```
kerberos health-check threshold milliseconds
```

where ***milliseconds*** (500-10,000) is the number of milliseconds to wait for the LDAP response.

This is a system-wide parameter; the same threshold applies to all DCs. The LDAP query is an unobtrusive method to test the DC's ability to perform lookups in the AD database.

For example, this command sets the health-check threshold to 3.5 seconds (3500 milliseconds):

```
bstnA(gbl)# kerberos health-check threshold 3500
bstnA(gbl)# . . .
```

Reverting to the Default Threshold

As mentioned above, the default health-check threshold for DCs is 2 seconds, or 2,000 milliseconds. You can use the no form of the kerberos health-check threshold command to return to this default:

```
no kerberos health-check threshold
```

For example, this reverts the "bstnA" chassis back to the default:

```
bstnA(gbl)# no kerberos health-check threshold
bstnA(gbl)# . . .
```

Showing DC Status

Use the show active-directory status command to see the status of the AD forest's DCs and all forest-to-forest trusts:

```
show active-directory status [forest forest-name | domain
domain-name] [detailed]
```

where you can use either of the options to focus on a single domain or forest:

forest-name (optional, 1-256 characters) shows one forest, or

domain-name (optional, 1-256 characters) focuses on one domain.

detailed (optional) expands the output so that it includes high-level statistics for each DC.

The output is a series of tables for each processor that runs the Kerberos software. The forest tables at the top of the output show which DC is active for each domain.

Each ARX processor makes independent decisions about which DC is active for a given domain. The output of this command shows separate status tables for each processor.

For example:

```
bstnA(gbl)# show active-directory status

Offline timeout is set to  3500 milliseconds.

PROCESSOR  1.1:

Forest :  MEDARCH.ORG
Domain Name                                         Domain Controller    Status    Preferred
--------------------------------------------------  -------------------  --------  ----
MEDARCH.ORG                                         192.168.25.102       Active    1
MEDARCH.ORG                                         192.168.25.104       Backup    1
MEDARCH.ORG                                         192.168.25.111       Backup    0
MEDARCH.ORG                                         192.168.25.109       Backup    0
MEDARCH.ORG                                         192.168.25.110       Backup    0
BOSTONMED.ORG                                       172.16.74.89         Active    0
BOSTONMED.ORG                                       172.16.74.88         Backup    0
FDTESTNET.NET                                       172.16.168.21        Active    0
FDTESTNET.NET                                       172.16.168.22        Backup    0
BOSTONCIFS.FDTESTNET.NET                            10.19.230.94         Active    0
BOSTONCIFS.FDTESTNET.NET                            10.19.230.88         Backup    0
WESTCOAST.MEDARCH.ORG                               192.168.202.10       Active    1
WESTCOAST.MEDARCH.ORG                               192.168.202.9        Backup    0
MA.NE.MEDARCH.ORG                                   192.168.25.103       Active    0
MA.NE.MEDARCH.ORG                                   192.168.25.105       Backup    0
NE.MEDARCH.ORG                                      172.16.124.73        Active    0
NE.MEDARCH.ORG                                      172.16.124.19        Backup    0

Forest :  NY.COM
Domain Name                                         Domain Controller    Status    Preferred
--------------------------------------------------  -------------------  --------  ----
NY.COM                                              172.16.108.136       Active    0
NY.COM                                              172.16.108.139       Backup    0
ADK.NY.COM                                          172.16.110.8         Active    0
ADK.NY.COM                                          172.16.110.5         Backup    0
CATSKILLS.NY.COM                                    172.16.120.22        Active    0
CATSKILLS.NY.COM                                    172.16.120.5         Backup    0

Forest :  VT.COM
Domain Name                                         Domain Controller    Status    Preferred
--------------------------------------------------  -------------------  --------  ----
VT.COM                                              172.16.213.9         Active    0
NH.ORG                                              172.16.210.14        Active    0
NH.ORG                                              172.16.210.7         Backup    0
MCNIELS.VT.COM                                      172.16.240.70        Active    0
MCNIELS.VT.COM                                      172.16.240.88        Backup    0


Forest Trust
```

```
    Forest-1             :  MEDARCH.ORG
    Forest-2             :  ny.com
    Trust Type           :  bidirectional
    Last Transition(UTC) :  05:54:19 02/24/2010
    Status               :  Forest roots are online

    Forest-1             :  ny.com
    Forest-2             :  vt.com
    Trust Type           :  bidirectional
    Last Transition(UTC) :  05:54:20 02/24/2010
    Status               :  Forest roots are online

bstnA(gbl)# ...
```

Removing an Active-Directory Forest

You can only remove an AD forest that does not contain any child domains or tree domains. You also cannot delete the forest if it is part of any forest-to-forest trusts (recall *Establishing Forest-to-Forest Trust*, on page 3-16). You can use show active-directory forest, described above, to confirm that the latest active-directory update removed all domains (except, possibly, the root domain) from the AD forest, and to confirm that the forest is not part of any forest-to-forest trusts.

From gbl mode, use the no active-directory-forest command to delete a forest configuration:

```
no active-directory-forest forest-name
```

where ***forest-name*** (1-256 characters) identifies the forest to remove.

For example, the following command removes the ‘testkerberos’ forest:

```
bstnA(gbl)# no active-directory-forest testkerberos
bstnA(gbl)# . . .
```

Authorizing Windows-Management (MMC/Snapshot) Access

You can define a group of Windows clients and their authority to use Windows-management tools, such as the MicroSoft Management Console (MMC). This group can use MMC and similar applications to view or edit CIFS shares, view and/or close open files, or view and/or close open client sessions. Additionally, you can allow or disallow access to snapshots for the group of Windows users. You can apply a Windows-management-authorization (*WMA*) group to any number of CIFS-supporting namespaces (as described in a later chapter).

◆ **Note**

If you have multiple CIFS namespaces associated with a single VIP, the WMA group must be the same for all of the namespaces.

Use the windows-mgmt-auth command to create a WMA group:

```
windows-mgmt-auth name
```

where ***name*** (1-64 characters) is a name you choose for the group.

This puts you into gbl-mgmt-auth mode, where you enter a list of Windows clients and the management access for them. Once the configuration is finished, you can apply it to any namespace in the clients' Windows domain.

For example, the following command sequence creates a WMA group called "readOnly:"

```
bstnA(gbl)# windows-mgmt-auth readOnly
bstnA(gbl-mgmt-auth[readOnly])# ...
```

Adding a Windows User to the Group

From gbl-mgmt-auth mode, use the user command to add one Windows client to the current WMA group:

```
user name windows-domain domain
```

where

name (1-64 characters) is the name of a valid Windows client, and

domain (1-64 characters) is the client's Windows domain. Use the exact domain name that the client uses for authentications. In a Kerberos environment, this is a fully-qualified-domain name (FQDN, such as "myco.mydiv.com"). In an NTLM environment, it may be a shorter NetBIOS name (such as "MYCO"). In a site that supports both Kerberos and NTLM, we suggest defining two user configurations for each client, one for each form of the domain (for example, "user john windows-domain myco.mydiv.com" followed by "user john windows-domain MYCO").

You can add multiple users to the group; invoke this command once (or possibly twice) for each user.

For example, the following command sequence adds five users to the WMA group, “readOnly:”

```
bstnA(gbl)# windows-mgmt-auth readOnly
bstnA(gbl-mgmt-auth[readOnly])# user mhoward_md windows-domain MEDARCH.ORG
bstnA(gbl-mgmt-auth[readOnly])# user zmarx_md windows-domain MEDARCH.ORG
bstnA(gbl-mgmt-auth[readOnly])# user lfine_md windows-domain MEDARCH.ORG
bstnA(gbl-mgmt-auth[readOnly])# user choward_md windows-domain MEDARCH.ORG
bstnA(gbl-mgmt-auth[readOnly])# user cjderita_md windows-domain MEDARCH.ORG
bstnA(gbl-mgmt-auth[readOnly])# ...
```

Removing a User

Use the no user command to remove one user from the current WMA group:

```
no user name windows-domain domain
```

where

name (1-64 characters) identifies the user, and

domain (1-64 characters) is the user’s Windows domain.

For example, the following command sequence removes one user from the “readOnly” group:

```
bstnA(gbl)# windows-mgmt-auth readOnly
bstnA(gbl-mgmt-auth[readOnly])# no user cjderita_md windows-domain MEDARCH.ORG
bstnA(gbl-mgmt-auth[readOnly])# ...
```

Setting Management Permissions for the Group

You can set read-only or read-write privileges for any of the following objects in a CIFS namespace:

- CIFS shares,
- CIFS-client sessions, and/or
- open files.

All users in the WMA group have the permissions you set with this command. By default, group members cannot change CIFS shares, client sessions, or open files, nor can they view potential CIFS shares (that is, non-shared volumes behind the CIFS service), client sessions, or open files. From gbl-mgmt-auth mode, use the permit command to enter one permission setting for the current WMA group:

```
permit {share | session | open-file | all} {monitor | any}
```

where

share | session | open-file | all chooses one type of object (or all of them), and

monitor | any chooses the permissions. The **any** flag allows group members to read and write the object(s); for example, **share any** means that group members can view, add, and delete CIFS shares from the namespace.

Re-use the command to enter more permission settings for this group.

The WMA group's members see the results of this command only if they connect after you invoke it.

◆ **Note**

Members with permit share any *permissions can remove shares only if they also have* permit session monitor *or* permit session any *permissions. This is because removing a share looks for an internal (non-user-visible) listing of all sessions that are connected to it.*

For example, the following command sequence permits the "readOnly" group to view (but not edit) CIFS shares and client sessions, then permits the group to view and/or close open files:

```
bstnA(gbl)# windows-mgmt-auth readOnly
bstnA(gbl-mgmt-auth[readOnly])# permit share monitor
bstnA(gbl-mgmt-auth[readOnly])# permit session monitor
bstnA(gbl-mgmt-auth[readOnly])# permit open-file any
bstnA(gbl-mgmt-auth[readOnly])# ...
```

Removing a Permission

You can remove access permissions for any or all of the CIFS objects (shares, client sessions, and/or open files). Use the no permit command to remove permissions from the current WMA group:

```
no permit {share | session | open-file | all}
```

where **share | session | open-file | all** chooses one type of object (or all of them).

For example, the following command sequence removes all open-file permissions from the "readOnly" group:

```
bstnA(gbl)# windows-mgmt-auth readOnly
bstnA(gbl-mgmt-auth[readOnly])# no permit open-file
bstnA(gbl-mgmt-auth[readOnly])# ...
```

Permitting Snapshot Access

A *snapshot* is an exact copy of a managed volume at a single point-in-time. You can create regularly-scheduled snapshots in a managed volume, and you can limit the CIFS clients who can access those snapshots. A WMA group defines the Windows users with snapshot access.

After you create a WMA group, use the permit snapshot monitor command to allow its users to view snapshots:

```
permit snapshot monitor
```

For example, the following command sequence creates a small “snapViewers” group and allows the group to access snapshots:

```
bstnA(gbl)# windows-mgmt-auth snapViewers
bstnA(gbl-mgmt-auth[snapViewers])# user juser windows-domain MEDARCH.ORG
bstnA(gbl-mgmt-auth[snapViewers])# user jquser windows-domain MEDARCH.ORG
bstnA(gbl-mgmt-auth[snapViewers])# permit snapshot monitor
bstnA(gbl-mgmt-auth[snapViewers])# ...
```

The *ARX CLI Maintenance Guide* contains detailed instructions on creating snapshots in a managed volume, and on using this type of WMA group to limit client access to snapshots.

Disabling Snapshot Access

Use the no form of the command to prevent members of the WMA group from accessing snapshots:

```
no permit snapshot monitor
```

For example, the following command sequence prevents the “readOnly” group from accessing snapshots:

```
bstnA(gbl)# windows-mgmt-auth readOnly
bstnA(gbl-mgmt-auth[readOnly])# no permit snapshot monitor
bstnA(gbl-mgmt-auth[readOnly])# ...
```

Showing All WMA Groups

You can use the show windows-mgmt-auth command to view all WMA groups on the ARX:

```
show windows-mgmt-auth
```

For example:

```
bstnA(gbl)# show windows-mgmt-auth

Windows Authorization Policy: fullAccess

User Name                                       Domain Name
----------------------------------------------- ------------------------------
juser                                           MEDARCH.ORG
jquser                                          MEDARCH.ORG

Managed Object          Permitted Operation
---------------------- ----------------------
All                     Any


Windows Authorization Policy: readOnly

User Name                                       Domain Name
----------------------------------------------- ------------------------------
mhoward_md                                      MEDARCH.ORG
zmarx_cpa                                       MEDARCH.ORG
lfine_md                                        MEDARCH.ORG
choward_md                                      MEDARCH.ORG

Managed Object          Permitted Operation
---------------------- ----------------------
```

```
Share                 Monitor
Session               Monitor
Snapshot              Monitor


Windows Authorization Policy: snapViewers

User Name                                      Domain Name
---------------------------------------------- -------------------------------
juser                                          MEDARCH.ORG
jquser                                         MEDARCH.ORG

Managed Object        Permitted Operation
--------------------- -----------------------
Snapshot              Monitor

bstnA(gbl)# ...
```

Focusing on One Group

To show a single WMA group, add the group name to the end of the show windows-mgmt-auth command:

show windows-mgmt-auth *name*

where ***name*** (1-64 characters) identifies the group to show.

For example:

```
bstnA(gbl)# show windows-mgmt-auth readOnly

Windows Authorization Policy: readOnly

User Name                                      Domain Name
---------------------------------------------- -------------------------------
mhoward_md                                     MEDARCH.ORG
zmarx_cpa                                      MEDARCH.ORG
lfine_md                                       MEDARCH.ORG
choward_md                                     MEDARCH.ORG

Managed Object        Permitted Operation
--------------------- -----------------------
Share                 Monitor
Session               Monitor
Snapshot              Monitor

bstnA(gbl)# ...
```

Removing a WMA Group

You can only remove a WMA group if it is not referenced by any namespace. A later chapter describes how to configure a namespace and reference a WMA group.

To remove a WMA group, use the no windows-mgmt-auth command in gbl mode:

no windows-mgmt-auth *name*

where ***name*** (1-64 characters) identifies the group to remove.

For example, the following command sequence removes a WMA group called “beta:”

```
bstnA(gbl)# no windows-mgmt-auth beta
bstnA(gbl)# ...
```

4

Preparing for NFS Authentication

Overview

You can create NFS access lists that filter clients based on their IP addresses. You can enter IP addresses directly and/or refer to pre-defined netgroups at a Network Information Service (NIS) server. A NIS *netgroup* defines a group of host machines, and may also contain other NIS netgroups.

This chapter pertains to NFS-client authentication only; you can skip this chapter unless you plan to offer NFS service with some level of client authentication.

Before You Begin

A NIS netgroup often refers to hosts by their DNS hostnames rather than their IP addresses. The ARX must use the local DNS server(s) to translate those names into IP addresses. Before you start configuring NIS domains, configure the switch to perform DNS lookups. Refer to the *ARX® CLI Network-Management Guide*: see *Configuring DNS Lookups*, on page 4-31.

All NIS servers used by the switch must also use the same DNS server(s). If this switch has a redundant peer, it must also use the same DNS server(s).

Adding a NIS Domain

The first step in using NIS netgroups is configuring a NIS domain on the switch. You can skip to the next section (*Adding an NFS Access List*, on page 4-9) if you do not plan to use any NIS netgroups in your NFS access lists.

The switch supports up to eight NIS domains. From gbl mode, use the nis domain command to add a new one:

```
nis domain domain
```

where ***domain*** (1-256 characters) is the name of the domain (for example, “f5” in “server.f5.com”). This must match the name of the NIS domain defined in one or more external NIS servers.

This places you into gbl-nis-dom mode, where you identify one or more NIS servers that host this NIS domain.

For example, the following command sequence adds a NIS domain named “wwmed.com:”

```
bstnA(gbl)# nis domain wwmed.com
bstnA(gbl-nis-dom[wwmed.com])# ...
```

Identifying a NIS Server for the NIS Domain

The final step in configuring an NIS domain is to identify one or more NIS servers that host the domain. From gbl-nis-dom mode, use the ip address command to identify one server:

```
ip address ip-address
```

where ***ip-address*** is in dotted-decimal format (for example, 192.168.25.122).

You can configure up to four NIS servers per NIS domain. They are used in the order that they are defined. If the first server times out or is unreachable, the ARX tries to connect to the next one, and so on.

For example, the following command identifies three NIS servers at 192.168.25.203, 204, and 205:

```
bstnA(gbl)# nis domain wwmed.com
bstnA(gbl-nis-dom[wwmed.com])# ip address 192.168.25.203
bstnA(gbl-nis-dom[wwmed.com])# ip address 192.168.25.204
bstnA(gbl-nis-dom[wwmed.com])# ip address 192.168.25.205
bstnA(gbl-nis-dom[wwmed.com])# ...
```

Removing a NIS Server

Use the no ip address command to remove one of the NIS servers from the list:

```
no ip address ip-address
```

where ***ip-address*** is in dotted-decimal format (for example, 192.168.25.122).

◆ **Important**

If you remove the only NIS server for the current NIS domain, support for the domain is limited. The switch keeps a local cache with all NIS netgroups, but can never refresh this cache. On the next switch reboot, the switch clears its NIS cache: this removes NIS support altogether. This can cause serious access issues for NFS clients that belong to the domain's netgroups.

For example, the following command removes one NIS server from the wwmed.com domain:

```
bstnA(gbl)# nis domain wwmed.com
bstnA(gbl-nis-dom[wwmed.com])# no ip address 192.168.25.205
bstnA(gbl-nis-dom[wwmed.com])# ...
```

Listing All Configured NIS Domains

You can use the show nis domain command to get a list of all configured NIS domains on the ARX:

```
show nis domain
```

For example, this command shows a single NIS domain backed by three servers:

```
bstnA(gbl)# show nis domain

NIS Domain                        Last Update   Status    Servers
--------------------------------  ------------  --------  ---------------
wwmed.com                         26 Jan 03:24  Success   192.168.25.201
                                                          192.168.25.204
                                                          192.168.25.205

bstnA(gbl)# ...
```

Showing Details for a NIS Domain

Add the name of an NIS domain to show details:

show nis domain *name*

where ***name*** (1-256 characters) identifies the NIS domain.

For example:

```
bstnA(gbl)# show nis domain wwmed.com

NIS Domain:                      wwmed.com
Server(s):                       192.168.25.201
                                 192.168.25.204
                                 192.168.25.205
Last Update:                     Fri Jan 26 03:24:16 2007
Last Update Status:              Success
Last Successful Update:          Fri Jan 26 03:24:16 2007
Netgroups:                       2397
Netgroup Resolution Errors:      0
Hosts:                           48046
Hosts Resolved:                  47507

bstnA(gbl)# ...
```

Listing Netgroups in a NIS Domain

You can use the show nis netgroup command to query the NIS-domain server for a list of the netgroups in the domain:

show nis netgroup *domain*

where ***domain*** (1-256 characters) identifies the NIS domain.

For example:

```
bstnA(gbl)# show nis netgroup wwmed.com

Netgroup
---------------------------------------------------------------------
auto_1
...
medtechs
surgeons

Total Netgroups: 2396
bstnA(gbl)# ...
```

Showing the Members of One Netgroup

For a list of members in a NIS netgroup, add the name of the netgroup to the end of the show nis netgroup command:

```
show nis netgroup domain netgroup
```

where

domain (1-256 characters) identifies the NIS domain, and

netgroup (1-1024 characters) is the specific netgroup.

This shows the host machines in the netgroup, along with their IP addresses. The ARX does not authenticate users and groups, so it does not use those IDs from the netgroup. The back-end filers authenticate users and groups, and the switch either allows or disallows client access based on the result.

For example, this shows the host machines in the "medtechs" netgroup:

```
bstnA(gbl)# show nis netgroup wwmed.com medtechs

Netgroup successfully resolved.

Hostname                                          IP Address
------------------------------------------------- -----------------
bench2.wwmed.com                                  10.51.201.72
bench3.wwmed.com                                  10.51.201.73
bench4.wwmed.com                                  10.51.201.74

Total Resolved Hosts: 3
bstnA(gbl)# ...
```

Updating the NIS Database

The ARX keeps an internal copy of all the NIS netgroups and their fully-resolved hosts. The database is built when you add the NIS domain to the switch; it is used for switch operation as well as the show commands above. To avoid excessive traffic to the DNS server, the switch does not update this database automatically. You can rebuild the database manually after any large-scale DNS or NIS changes.

From priv-exec mode, use the nis update command to update the netgroup database:

```
nis update [nis-domain]
```

where ***nis-domain*** (optional: 1-256 characters) causes the update to focus on a single NIS domain.

In a redundant pair of switches, this triggers concurrent NIS updates from both peers. Since they keep independent NIS caches, failovers do not incur any additional downtime for NIS.

The show nis domain command (above) shows the time of the most-recent update.

For example, the following command sequence updates all NIS domains:

```
bstnA(gbl)# end
bstnA# nis update
bstnA# ...
```

Reading the Update Report

Each NIS update creates one or more reports (one per updated domain) to show its results. The NIS software names the reports according to the following convention:

nis-update.*domain-name*.rpt

where *domain-name* is the name of the NIS domain in the report.

To conserve disk space, each NIS update overwrites any previous reports.

Use the show reports type NIS command to list all NIS-update reports.

```
show reports type NIS
```

Use show reports *report-name*, tail, or grep to read the file. To save the report off to an external FTP site, use the copy ... ftp command from priv-exec mode. To upload the report to an SCP host, use copy ... scp. All of these commands are documented in the *CLI Reference* manual.

This report lists all the hosts in the NIS domain that had issues, such as the name not being found at the DNS server. You can use this report as a guide to adjust the DNS and/or NIS configurations on the back-end servers.

For example, this shows a NIS-update report for a large domain, “wwmed.com:”

```
bstnA(gbl)# show reports nis-update.wwmed.com.rpt
**** NIS Update Report: Started at Mon Sep 24 03:54:04 2007 ****
**** Software Version: 3.00.000.10541 (Sep 21 2007 18:11:56) [nbuilds]
**** Hardware Platform: ARX-6000
**** NIS Domain: wwmed.com
**** NIS Server: 192.168.25.201
**** NIS Server: 192.168.25.204
**** NIS Server: 192.168.25.205

**** Legend:
****   HN = Hostname not found.
****   NP = Netgroup parsing error.
****   NG = Netgroup not found.
****   WG = Watched netgroup has changed contents.
****   NE = NIS server error.

Status            Hostname/Netgroup
----------------  ------------------------------------------------------------
[HN            ]  not_a_real_host1 in group: bad_hosts
[HN            ]  not_a_real_host2 in group: bad_hosts
[HN            ]  not_a_real_host3 in group: bad_hosts
[HN            ]  not_a_real_host4 in group: bad_hosts
[HN            ]  not_a_real_host5 in group: bad_hosts
[HN            ]  baghdad in group: chassis
[HN            ]  ommegang in group: chassis
[HN            ]  bismark in group: chassis

...

[HN            ]  london in group: sixthousands
[HN            ]  montreal in group: sixthousands
[HN            ]  lasvegas in group: sixthousands

Netgroups Processed:                        2,396
Hosts Processed:                           48,043
```

```
Hostnames Not Found:                         539
Netgroup Parsing Errors:                       0
Netgroups Not Found:                           0
Watched Netgroup Changes                       0

**** Elapsed time:           00:00:17
**** NIS Update Report: DONE at Wed Dec  7 09:45:20 2005 ****
bstnA(gbl)# ...
```

Scheduling Regular Updates

If your site makes smaller, incremental changes to DNS and/or NIS netgroups, you can use the at command to arrange for regular NIS updates. The at command is in cfg mode.

As mentioned above, a redundant pair keeps independent NIS caches on each peer. Schedule these updates on both peers.

For example, the following command sequence runs the nis update command every night at midnight:

```
bstnA(gbl)# end
bstnA# config
bstnA(cfg)# at 12:00:00 every 1 days do “nis update”

  The scheduled execution time for CLI command is: 8/30/07 12:00 PM.

bstnA(cfg)# ...
```

As another example, the following command sequence causes nis update to run every Saturday at midnight:

```
bstnA(gbl)# end
bstnA# config
bstnA(cfg)# at 12:00:00 every saturday do “nis update”

  The scheduled execution time for CLI command is: 9/01/07 12:00 PM.

bstnA(cfg)# ...
```

Removing the NIS Domain-Server Map

From gbl mode, use no nis domain to remove a NIS domain-server map:

no nis domain *domain*

where ***domain*** (1-256 characters) is the name of the domain to remove.

You cannot remove a domain that is referenced by an NFS access list. The next section describes how to use an NIS domain in an access list.

For example, the following command sequence removes the NIS domain named “PROVIDENCE:”

```
bstnA(gbl)# no nis domain PROVIDENCE
bstnA(gbl)# ...
```

Adding an NFS Access List

Before you configure any NFS exports or services, you can optionally configure one or more NFS access lists. An *access list* is a list of subnets and hosts to which you permit or deny access to NFS service. For example, you could permit access from the subnet at 192.168.101.0 but deny access from all other subnets.

◆ **Note**

If you currently use NIS at your back-end filers, the front-end NFS service passes the client's User ID and Group ID through to the back-end filer, and the back-end filer authenticates against those IDs as usual. The access list filters out users based on IP address only, but the filer(s) may deny access based on user ID after the IP address passes at the ARX.

When you configure NFS shares and services later, you can apply one NFS access list to each NFS share. You can reuse one NFS access list for any number of NFS shares.

From gbl mode, use the nfs-access-list command to create a new one:

```
nfs-access-list list-name
```

where ***list-name*** (1-64 characters) is a name you choose for the access list.

The CLI prompts for confirmation before creating the new NFS access list. Enter **yes** to proceed. This places you in gbl-nfs-acl mode, from which you can configure various permit and deny rules for specific subnets and/or NIS netgroups. By default, all subnets and netgroups are denied any access.

For example, the following command sequence creates a new NFS access list:

```
bstnA(gbl)# nfs-access-list eastcoast

This will create a new NFS ACL.

Create NFS ACL ''eastcoast''? [yes/no] yes
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Listing All NFS Access Lists

To verify that your NFS access list was added to the configuration, use the show nfs-access-list command. You can invoke this command from any mode.

```
show nfs-access-list
```

The output shows all configured access lists in a table, one access list per row. For each access list, high-level details are shown. These details are clarified in the sections below.

For example:

```
bstnA(gbl)# show nfs-access-list
Access List Name    Anon UID Anon GID    Num Rules      Num References
```

```
  ----------------------------------------------------------------------
  eastcoast          100      100              8                   2
  westcoast          65534    65534            2                   0
bstnA(gbl)# ...
```

Showing One NFS Access List

As you configure your NFS access lists, it will be convenient to see the current list settings. Use the show nfs-access-list command with the specific access-list name to see the full configuration for one access list:

```
show nfs-access-list list-name
```

where ***list-name*** (1-64 characters) identifies the access list.

Among other configuration details, the output shows the order of all permit and deny rules in the access list. As explained below, the order of these rules is important.

For example:

```
bstnA(gbl)# show nfs-access-list eastcoast

Access List Name: eastcoast
  Description:              allowable subnets in MA, NY, & DC
  NIS Domain:               wwmed.com
  Anonymous UID:            100
  Anonymous GID:            100
  Number of References:     2

permit 172.16.100.0 255.255.255.0 read-write root squash
permit 172.16.204.0 255.255.255.0 read-only  root allow
permit 172.16.0.0 255.255.0.0 read-write root squash
permit netgroup surgeons read-write root allow
permit netgroup medtechs read-only  root squash
deny  192.168.77.0 255.255.255.0
deny  192.168.202.0 255.255.255.0
permit 192.168.0.0 255.255.0.0 read-write root squash

bstnA(gbl)# ...
```

Resolving All Netgroups in the Access List

If the access list contains any netgroups, you can resolve those netgroups to see all of the hosts within them. To accomplish this, add the resolve-netgroups keyword to the end of the command:

```
show nfs-access-list list-name resolve-netgroups
```

This provides a complete view of the access list, resolving all netgroups to the IP addresses for their hosts. For example, this expands the previous output to show all hosts in the “surgeons” and “medtechs” netgroups:

```
bstnA(gbl)# show nfs-access-list eastcoast resolve-netgroups

Access List Name: eastcoast
  Description:              allowable subnets in MA, NY, & DC
  NIS Domain:               wwmed.com
  Anonymous UID:            100
  Anonymous GID:            100
  Number of References:     2
```

```
permit 172.16.100.0 255.255.255.0 read-write root squash
permit 172.16.204.0 255.255.255.0 read-only  root allow
permit 172.16.0.0 255.255.0.0 read-write root squash
permit 10.51.201.71 255.255.255.255 read-write root allow
permit 10.51.201.72 255.255.255.255 read-only  root squash
permit 10.51.201.73 255.255.255.255 read-only  root squash
permit 10.51.201.74 255.255.255.255 read-only  root squash
deny  192.168.77.0 255.255.255.0
deny  192.168.202.0 255.255.255.0
permit 192.168.0.0 255.255.0.0 read-write root squash

Number of entries in access list: 10
All Netgroup(s) were successfully resolved.

bstnA(gbl)# ...
```

Each access list can support a maximum of 2048 permit and deny rules, including the individual permit rules for every host in every netgroup. If you exceed the limit (perhaps because of an overly-large netgroup), this output shows the first 2048 entries followed by an error.

Setting the NIS Domain

If the access list will use NIS netgroups, you must set the access list's NIS domain. The ARX needs the NIS domain to access the local NIS server. (You map the NIS domain name to an NIS server ahead of time, as described earlier: recall *Adding a NIS Domain*, on page 4-3.)

From gbl-nfs-acl mode, use the nis domain command to set the NIS domain for the access list:

nis domain *domain*

where ***domain*** can be up to 256 characters long.

For example, the following command sequence lists all NIS domains, shows only "wwmed.com," and uses it in an access list named "eastcoast:"

```
bstnA(gbl)# show nis domain

NIS Domain                        Last Update   Status    Servers
--------------------------------  ------------  --------  ---------------
wwmed.com                         26 Jan 03:24  Success   192.168.25.201
                                                          192.168.25.204
                                                          192.168.25.205

bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# nis domain wwmed.com
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Removing the NIS Domain

Use no nis domain to remove the NIS domain from the access list:

no nis domain

For example:

```
bstnA(gbl)# nfs-access-list westcoast
bstnA(gbl-nfs-acl[eastcoast])# no nis domain snemed.com
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Adding a Permit Rule

By default, a new NFS access list denies access to all subnets. You can selectively allow access by configuring a *permit* rule for each trusted subnet. From gbl-nfs-acl mode, use the permit command to add a permit rule for one subnet:

```
permit ip-address mask [read-only]
```

where

ip-address is the address of the subnet,

mask defines the network part of the *ip-address*, and

read-only is an optional flag to permit read access but deny writes. If omitted, this defaults to allowing both reads and writes.

For example, the following command sequence permits read-write access to clients at 172.16.100.0:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# permit 172.16.100.0 255.255.255.0
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Permitting a Netgroup

If you have configured a NIS domain for this access list (see above), you can refer to a netgroup configured in that domain. This leverages any netgroups that were configured before the introduction of the ARX. From gbl-nfs-acl mode, use the permit netgroup command to permit a NIS netgroup:

```
permit netgroup name [read-only]
```

where

name (1-1024 characters) identifies a netgroup from the NIS domain, and

read-only is an optional flag to permit read access but deny writes. If omitted, this defaults to allowing both reads and writes.

This permits all hosts in the netgroup. As explained earlier, the netgroups users and/or groups are handled at the back-end filer(s), not the switch.

For example, the following command sequence shows all netgroups in the “wwmed.com” domain and permits read-only access to host machines in the “medtechs” netgroup:

```
bstnA(gbl)# show nis netgroup wwmed.com

Netgroup
------------------------------------------------------------------
auto_1
...
medtechs
surgeons

Total Netgroups: 2396
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# permit netgroup medtechs read-only
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Rule Ordering

The order of rules is very important in an access list. Whenever a client tries to access an NFS service with an access list, the client's IP address is compared to the rules in the order they were entered. If the IP address matches two rules, the first rule is used and the second rule is ignored.

For example, consider the two permit rules below. Clients in 192.168.10.x would match the first rule, while clients outside that subnet in the same Class-B network (192.168.x.x) would match the second rule. This would give read-only access to clients in the Class-B network but full read-write access clients in the smaller Class-C subnet:

```
permit 192.168.10.0 255.255.255.0
permit 192.168.0.0 255.255.0.0 read-only
```

If the rules were reversed, clients in the Class-C subnet would match the read-only rule before reaching the read-write rule that was intended for them.

Allowing Root Access

A new permit rule squashes root access by default. That is, if a client logs in as the *root* user (sometimes called the superuser) and accesses the NFS share, the ARX translates the client's user ID to an *anonymous* ID with very low access privileges. The client can therefore write only to files or directories with wide-open permission settings. This is the safest strategy for a permit rule, as it prevents *root* users from damaging NFS shares.

You have the option to disable root squashing in a permit rule. From gbl-nfs-acl mode, use the root allow keywords at the end of the permit command to allow root access:

permit *ip-address mask* [read-only] root allow

or

permit netgroup *name* [read-only] root allow

◆ **Important**

This setting permits clients with root access to change or remove any (or all) files or directories. Whether by accident or malicious intent, this could result in loss or corruption in client data.

For example, the following command sequence allows *root* access from clients at 172.16.204.0. To control the security problem, access is read-only for this rule:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# permit 172.16.204.0 255.255.255.0 read-only root allow
bstnA(gbl-nfs-acl[eastcoast])# ...
```

As another example, this command sequence allows *root* access from clients in the "surgeons" netgroup:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# permit netgroup surgeons root allow
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Removing a Permit Rule

From gbl-nfs-acl mode, use no permit to remove a permit rule from the current access list:

no permit *ip-address mask*

where:

ip-address identifies the subnet for the permit rule, and

mask defines the network part of the *ip-address*.

or

no permit netgroup *name*

where ***name*** (1-1024 characters) identifies the netgroup to remove.

For example, the following command sequence removes the permit rule for clients at 172.16.13.0:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# no permit 172.16.13.0 255.255.255.0
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Changing the Anonymous User ID

When permit rules have root-squash enabled, they translate the User ID (UID) of a *root* user to an *anonymous* UID. By default, the access list uses 65534 for this UID. To change the UID for *anonymous*, use the anonymous-uid command:

anonymous-uid *id*

where ***id*** is a number from 1-65535.

For example, the following command sequence sets the *anonymous* UID to 100:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# anonymous-uid 100
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Changing the Anonymous Group ID

When permit rules have root-squash enabled, they translate the Group ID (GID) of a *root* user to an *anonymous* GID. By default, the access list uses 65534. To change the GID for *anonymous*, use the anonymous-gid command:

anonymous-gid *id*

where ***id*** is a number from 1-65535.

For example, the following command sequence sets the *anonymous* GID to 100:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# anonymous-gid 100
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Reverting to the Default Group ID

For root squashing, an access list uses the default GID of 65534. To revert to this default, use the no anonymous-gid command:

```
no anonymous-gid
```

For example:

```
bstnA(gbl)# nfs-access-list westcoast
bstnA(gbl-nfs-acl[westcoast])# no anonymous-gid
bstnA(gbl-nfs-acl[westcoast])# ...
```

Reverting to the Default User ID

As with the GID, an access list uses the default UID of 65534 when it performs root squashing. From gbl-nfs-acl mode, use the no anonymous-uid command to revert to this default:

```
no anonymous-uid
```

For example:

```
bstnA(gbl)# nfs-access-list westcoast
bstnA(gbl-nfs-acl[westcoast])# no anonymous-uid
bstnA(gbl-nfs-acl[westcoast])# ...
```

Adding a Deny Rule

You may have a situation where most of a large subnet should be permitted access to NFS, but some portions of the subnet should be denied access. From gbl-nfs-acl mode, use the deny command to add a deny rule for one subnet:

```
deny ip-address mask
```

where

ip-address is the address of the subnet, and

mask defines the network part of the *ip-address*.

For example, the following command sequence denies access to two Class-C subnets, but then permits access to any IP outside those subnets but *inside* their Class-B supernet, 192.168.0.0/16:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# deny 192.168.77.0 255.255.255.0
bstnA(gbl-nfs-acl[eastcoast])# deny 192.168.202.0 255.255.255.0
bstnA(gbl-nfs-acl[eastcoast])# permit 192.168.0.0 255.255.0.0
bstnA(gbl-nfs-acl[eastcoast])# ...
```

You cannot deny a NIS netgroup. We recommend a subnet-deny rule after any permit netgroup rule, to ensure that all other hosts in the netgroup’s subnet are explicitly denied.

Removing a Deny Rule

From gbl-nfs-acl mode, use no deny to remove a deny rule from the current access list:

no deny *ip-address mask*

where

ip-address identifies the subnet for the deny rule, and

mask defines the network part of the *ip-address*.

For example, the following command sequence removes the deny rule for clients at 192.168.77.0:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# no deny 192.168.77.0 255.255.255.0
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Changing Rule Order

The order that you enter rules determines the order in which they are compared to client IPs. If a client’s IP address matches more than one rule in the access list, the ARX uses the first matching rule and ignores the rest. For example, consider the following access list:

```
bstnA(gbl)# show nfs-access-list eastcoast

...
permit 192.168.0.0 255.255.0.0 read-write root squash
deny  192.168.77.0 255.255.255.0
deny  192.168.202.0 255.255.255.0
...
```

These permit and deny rules have a subtle configuration error. The intention was to allow all clients from 192.168.0.0 *except* clients from 192.168.77.0 or 192.168.202.0. For example, a client at IP 192.168.77.29 is supposed to be blocked by the first deny rule, “deny 192.168.77.0 ...” However, that IP address matches the Class-B network (192.168.0.0) in the earlier permit rule. The deny rules can never actually be reached.

To correct this configuration error, you must delete the rule(s) that is/are out of order, then add rules back into the access list in the correct order. This re-ordering method conforms to industry standards for configuring access lists.

The following command sequence shows an NFS access list with this mis-configuration and then corrects it:

```
bstnA(gbl)# show nfs-access-list eastcoast

Access List Name: eastcoast
  Description:              allowable subnets in MA, NY, & DC
  NIS Domain:               wwmed.com
  Anonymous UID:            100
  Anonymous GID:            100
  Number of References:     1

permit 172.16.100.0 255.255.255.0 read-write root squash
permit 172.16.204.0 255.255.255.0 read-only  root allow
permit 172.16.0.0 255.255.0.0 read-write root squash
permit netgroup surgeons read-write root allow
permit netgroup medtechs read-only  root squash
permit 192.168.0.0 255.255.0.0 read-write root squash
deny  192.168.77.0 255.255.255.0
deny  192.168.202.0 255.255.255.0
```

First, remove the permit rule and show that it is gone:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# no permit 192.168.0.0 255.255.0.0
bstnA(gbl-nfs-acl[eastcoast])# show nfs-access-list eastcoast
...
permit netgroup surgeons read-write root allow
permit netgroup medtechs read-only  root squash
deny  192.168.77.0 255.255.255.0
deny  192.168.202.0 255.255.255.0
```

Add back the permit rule and show that it is now at the end of the list:

```
bstnA(gbl-nfs-acl[eastcoast])# permit 192.168.0.0 255.255.0.0
bstnA(gbl-nfs-acl[eastcoast])# show nfs-access-list eastcoast
...
deny  192.168.77.0 255.255.255.0
deny  192.168.202.0 255.255.255.0
permit 192.168.0.0 255.255.0.0 read-write root squash
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Exit the Mode to Enforce Any Permit/Deny Rules

You must leave gbl-nfs-acl mode to enforce its permit or deny rules. This ensures that the access list assesses all of the rules at once, in order, and does not enforce an access list in a transitional state. Use the **exit** or **end** command to leave the mode.

Adding a Description

You can add a description to the access list for use in the show command. The description can differentiate the list from others. From gbl-nfs-acl mode, use the description command to add a description:

```
description text
```

where ***text*** is 1-255 characters. Quote the text if it contains any spaces.

For example:

```
bstnA(gbl)# nfs-access-list eastcoast
bstnA(gbl-nfs-acl[eastcoast])# description “allowable subnets in MA, NY, & DC”
bstnA(gbl-nfs-acl[eastcoast])# ...
```

Removing the Description

From gbl-nfs-acl mode, use no description to remove the description string from the current access list:

```
no description
```

For example:

```
bstnA(gbl)# nfs-access-list westcoast
bstnA(gbl-nfs-acl[westcoast])# no description
bstnA(gbl-nfs-acl[westcoast])# ...
```

Removing an Access List

From gbl mode, use the no nfs-access-list command to remove an NFS access list:

no nfs-access-list *list-name*

where ***list-name*** (1-64 characters) identifies the access list to remove.

You must remove all references to the access list before you can use this command to remove the list itself. An access list is referenced from an NFS service; instructions for configuring an NFS service appear below.

For example, the following command sequence removes the NFS access list, “southwest:”

```
bstnA(gbl)# no nfs-access-list southwest
bstnA(gbl)# ...
```

5

Examining Filers

Overview

Use the show exports and probe exports commands to examine filers in the server network. These commands make queries from proxy-IP addresses to test filer connectivity, find the services supported by the filer, discover filer shares, and discover permissions settings at the shares. You can use them to troubleshoot network connectivity to filers as well as any permissions issues.

From any mode, use show exports to examine a filer's shares and capabilities:

```
show exports host filer
```

where ***filer*** (1-1024 characters) is the filer's IP address or host name. You can use the hostname only if the switch is configured for DNS lookups; refer to *Configuring DNS Lookups*, on page 4-31 in the *ARX® CLI Network-Management Guide*.

This shows a report with five or more tables:

- Connectivity - shows whether or not pings succeed from every active proxy-IP address. All proxy-IP pings must succeed. If they fail, verify that the filer is on the proxy-IP subnet (*Adding a Range of Proxy-IP Addresses*, on page 4-7 of the *ARX® CLI Network-Management Guide*) or reachable through a gateway on that subnet (via static route: see *Adding a Static Route*, on page 4-10 of the network guide). If some succeed and some fail for a given proxy IP, the MTU settings are probably set inconsistently between the proxy IP, the filer, and some of the network equipment between them. For a command to change the MTU on an ARX VLAN, refer to *Enabling Jumbo Frames (optional)*, on page 3-8 of the network guide.
- CIFS Credentials - only appears for a filer that supports CIFS. This shows the CIFS username and Windows domain that the show exports command used to authenticate. If the domain is an FQDN, the command attempts to use Kerberos authentication. You have options to choose these credentials, described later in this chapter.
- Capabilities - shows the transport protocols (TCP or UDP) and port numbers for NFS and CIFS. For NFS, this shows the same information for portmapper and the mount daemon: an NFS filer must support all three services (portmapper, mount, and NFS itself). For CIFS servers, this also shows the SMB and/or SMB2 capabilities, supported authentication protocols, whether or not it can connect to the IPC$ share, and information about the server's SPN.
- Shares - For NFS, this shows the Path of the export, the results of a UNIX **showmount -e** command (which may be blank, depending on the text string that the filer returns), and the Status of a test mount from the ARX. The test mount must succeed for the share to be usable in a namespace.

For CIFS, this shows each Share name, the Total and Free storage space on the share, and the Serial Num of the storage volume behind the share. Note that the space measures apply to the storage volume behind the share; if multiple shares map to the same serial number (and therefore the same storage volume), their space measures are all the same.

- Time - this shows the time skew between the ARX and the filer, if any. Namespace policy and Kerberos authentication (described in later chapters) require that the clocks be nearly synchronized between the ARX and its filers; NTP is recommended. If the filer has both CIFS and NFS shares, there may be different skews for each server.

If a show exports CLI command specifies a user or proxy-user, the command returns only CIFS information for the specified filer. If the command does not specify a user or proxy-user, the command returns only NFS information for the specified filer.

For example, the following command shows all of the tables described above for a filer at 192.168.25.19. This command does not include any user credentials, and therefore does not return any information about the filer's CIFS shares. (The next section explains how to supply CIFS credentials with this command.)

```
bstnA# show exports host 192.168.25.19
Export probe of filer "192.168.25.19"

Connectivity:

  Slot.Proc  Proxy Address    Ping (size: result)
  ---------  ---------------  -------------------
  2.1        192.168.25.31    64: Success  2000: Success  8820: Success
  2.2        192.168.25.32    64: Success  2000: Success  8820: Success
  2.3        192.168.25.33    64: Success  2000: Success  8820: Success
  2.4        192.168.25.34    64: Success  2000: Success  8820: Success
  2.5        192.168.25.141   64: Success  2000: Success  8820: Success
  2.6        192.168.25.142   64: Success  2000: Success  8820: Success
  2.7        192.168.25.143   64: Success  2000: Success  8820: Success
  2.8        192.168.25.144   64: Success  2000: Success  8820: Success
  2.9        192.168.25.145   64: Success  2000: Success  8820: Success
  2.10       192.168.25.146   64: Success  2000: Success  8820: Success
  2.11       192.168.25.147   64: Success  2000: Success  8820: Success
  2.12       192.168.25.148   64: Success  2000: Success  8820: Success

Capabilities:
  NFS
    Port Mapper    TCP/111, UDP/111
    Mount Daemon   V1 TCP/959, V1 UDP/956, V2 TCP/959, V2 UDP/956, V3 TCP/959, V3 UDP/956
    Server         V2 UDP/2049, V3 TCP/2049, V3 UDP/2049

Shares:
  NFS
                                                  Read   Write     Space
    Path (Owner)                         Status   Size   Size   Total  Avail   FSID   Access
    --------------------------------     ------- ------ ------ ------ ------ -------- ------
    /exports                             Mounted 8.0 kB 8.0 kB  89 GB  81 GB 809      *
    /exports/share_10-nfs-ARXMax-FOXX    192.168.25.2: FILER_MOUNT_FAILED_NOENT *
    /exports/share_1002-nfs-ARXMax-FOXX  192.168.25.2: FILER_MOUNT_FAILED_NOENT *
    /exports/share_1006-nfs-ARXMax-FOXX  192.168.25.2: FILER_MOUNT_FAILED_NOENT *
    /exports/share_1010-nfs-ARXMax-FOXX  192.168.25.2: FILER_MOUNT_FAILED_NOENT *
    /exports/share_1014-nfs-ARXMax-FOXX  192.168.25.2: FILER_MOUNT_FAILED_NOENT *
    /exports/share_1018-nfs-ARXMax-FOXX  192.168.25.2: FILER_MOUNT_FAILED_NOENT *
    /exports/share_102-nfs-ARXMax-FOXX   192.168.25.2: FILER_MOUNT_FAILED_NOENT *
    .
    .
    .
    /exports/share_990-nfs-ARXMax-FOXX   192.168.25.2: FILER_MOUNT_FAILED_NOENT *
    /exports/share_994-nfs-ARXMax-FOXX   192.168.25.2: FILER_MOUNT_FAILED_NOENT *
    /exports/share_998-nfs-ARXMax-FOXX   192.168.25.2: FILER_MOUNT_FAILED_NOENT *

Time:
  NFS
    Filer's time is the same as the switch's time.
bstnA#
```

Examining CIFS Shares

You can only examine CIFS shares if you have sufficient permissions at the filer. Use the user and windows-domain options to provide Windows credentials to the filer:

```
show exports host filer user username windows-domain domain
```

where

username (1-32 characters) is the username.

domain (1-1024 characters) is the user’s Windows domain. Use an FQDN (for example, “ourco.parentco.com” instead of “ourco”) if you want to authenticate with Kerberos.

The other options are explained above.

The CLI prompts for the user’s password. Enter the password to continue. The command only shows CIFS shares, capabilities, and time.

Optionally, after you specify the filer name or address in the command, you can use the spn argument to specify a service principal name for identifying the filer.

◆ **Note**

The filer queries can take more than one minute for a filer with a large number of CIFS shares.

For example, this is a CIFS-side probe of a filer at 192.168.25.21.

```
bstnA# show exports host 192.168.25.21 user jqpublic windows-domain MEDARCH.ORG
Password: ******
Export probe of filer "192.168.25.21"

Connectivity:

  Slot.Proc  Proxy Address    Ping (size: result)
  ---------  ---------------  -------------------
  2.1        192.168.25.31    64: Success  2000: Success  8820: Success
  2.2        192.168.25.32    64: Success  2000: Success  8820: Success
  2.3        192.168.25.33    64: Success  2000: Success  8820: Success
  2.4        192.168.25.34    64: Success  2000: Success  8820: Success
  2.5        192.168.25.141   64: Success  2000: Success  8820: Success
  2.6        192.168.25.142   64: Success  2000: Success  8820: Success
  2.7        192.168.25.143   64: Success  2000: Success  8820: Success
  2.8        192.168.25.144   64: Success  2000: Success  8820: Success
  2.9        192.168.25.145   64: Success  2000: Success  8820: Success
  2.10       192.168.25.146   64: Success  2000: Success  8820: Success
  2.11       192.168.25.147   64: Success  2000: Success  8820: Success
  2.12       192.168.25.148   64: Success  2000: Success  8820: Success

CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG

Capabilities:

  CIFS
    Server            TCP/445, TCP/139
    SMB:
      Security Mode   User level, Challenge/response, Signatures disabled
      Capabilities    Large Files, Large Read, Large Write
      Max Buffer      33028
    Auth Protocols    NTLMSSP, Kerberos
    IPC$ Share        Access OK
    Auth Method Used  Kerberos
    SPN Used          ntap820$@MEDARCH.ORG
    Discovered SPN    ntap820$@MEDARCH.ORG

Shares:

  CIFS
                                                  Space
    Share                                   Total        Avail      Serial Num
    -------------------------------    ----------   ----------      ----------
    ADAM_ft_cifs_al_1                       56 GB        54 GB       ae01-8ca5
    ADAM_ft_cifs_al_2                       56 GB        54 GB       ae01-8ca5
    ADAM_ft_co_ds_v1_s1                     56 GB        54 GB       ae01-8ca5
    .
    .
    .
    vol1snuppy                              63 GB        58 GB       480d-8ca5
    vol2snuppy                              17 GB        16 GB       a808-8ca5
    vol3snuppy                              68 GB       3.8 MB       a702-8ca5
    vol4snuppy                              68 GB        26 GB       a502-8ca5
    vol5snuppy                              68 GB        39 GB       a902-8ca5
    volvol2                                 17 GB        16 GB       a801-8ca5
    -------------------------------    ----------   ----------      ----------
    Share                                   Total        Avail      Serial Num
                                                  Space

Time:

  CIFS
    Filer's time is the same as the switch's time.
bstnA# ...
```

Using Proxy-User Credentials

If there is a proxy user that is already configured for the filer’s domain, you can use the proxy user configuration instead of a username, domain, and password. Use show proxy-user for a full list of all proxy users (recall *Listing All Proxy Users*, on page 3-7). The CLI does not prompt for a password if you use a pre-configured proxy user:

show exports host *filer* proxy-user *proxy*

where

proxy (1-32 characters) is the name of the proxy-user configuration, and

the other options are explained above.

As with the user/windows-domain syntax, this limits the output of the command to CIFS shares, capabilities, and time readings.

For example, this command sequence uses a set of proxy-user credentials to examine the filer at 192.168.25.20:

```
bstnA> show proxy-user

Name             Windows Domain                Pre-Win2k        User Name
                 Description
--------------------------------------------------------------------------------
acoProxy1        WWMEDNET.COM                  WWMEDNET         jqprivate
                 jq's admin account

acoProxy3        FDTESTNET.COM                 BOSTONCIFS       jqtester

cifs_admin       MEDARCH.ORG                   MEDARCH          Administrator

nas_admin                                                       root

ny_admin         NY.COM                        NY               jqpublic

acoProxy2        MEDARCH.ORG                   MEDARCH          jqpublic
                 user with backup and admin creds on our servers

bstnA# show exports host 192.168.25.20 proxy-user acoProxy2
Export probe of filer "10.51.200.20"

Connectivity:

  Slot.Proc  Proxy Address    Ping (size: result)
  ---------  ---------------  -------------------
  2.1        192.168.25.31    64: Success  2000: Success  8820: Success
  2.2        192.168.25.32    64: Success  2000: Success  8820: Success
  2.3        192.168.25.33    64: Success  2000: Success  8820: Success
  2.4        192.168.25.34    64: Success  2000: Success  8820: Success
  2.5        192.168.25.141   64: Success  2000: Success  8820: Success
  2.6        192.168.25.142   64: Success  2000: Success  8820: Success
  2.7        192.168.25.143   64: Success  2000: Success  8820: Success
  2.8        192.168.25.144   64: Success  2000: Success  8820: Success
  2.9        192.168.25.145   64: Success  2000: Success  8820: Success
  2.10       192.168.25.146   64: Success  2000: Success  8820: Success
  2.11       192.168.25.147   64: Success  2000: Success  8820: Success
  2.12       192.168.25.148   64: Success  2000: Success  8820: Success
```

```
CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG
    Pre-Win2k Domain  MEDARCH

Capabilities:

  CIFS
    Server            TCP/445
    SMB:
      Security Mode   User level, Challenge/response, Signatures disabled
      Capabilities    Large Files, Large Read, Large Write, Info Passthru
      Max Buffer      16644
    SMB2.1:
      Security Mode   Signatures optional
      Capabilities    Leasing, Large MTU
      Max Read        1048576
      Max Write       1048576
      Max Transact    1048576
    Auth Protocols    NTLMSSP, Kerberos
    IPC$ Share        Access OK
    Auth Method Used  Kerberos
    SPN Used          VM-SMB2K8R2-01@MEDARCH.ORG
    Discovered SPN    VM-SMB2K8R2-01@MEDARCH.ORG

Shares:

  CIFS
                                                  Space
    Share                                   Total        Avail      Serial Num
    -------------------------------    ----------   ----------      ----------
    DREW_1                                  29 GB        25 GB       1473-165b
    DREW_4                                  29 GB        25 GB       1473-165b
...
    yamaha_es_cifs1                         29 GB        25 GB       1473-165b
    yamaha_es_cifs2                         29 GB        25 GB       1473-165b
    -------------------------------    ----------   ----------      ----------
    Share                                   Total        Avail      Serial Num
                                                  Space

Time:

  CIFS
    Filer's time is the same as the switch's time.
bstnA# ...
```

Showing the Physical Paths for CIFS Shares

For the physical disk and path behind each CIFS share, use the optional paths keyword after the filer hostname/IP:

show exports host *filer* paths
[user *username* windows-domain *domain* | proxy-user *proxy*]

where the options are explained above.

This shows the relationships between shares. If a share is inside the directory tree of another share, it is called a *subshare* of its parent share. You can use this command to identify all share-to-subshare relationships on the filer.

◆ **Note**

To read the shares' paths at the back-end filers, the volume requires Windows credentials that belong to the Administrators group on each filer. This is a significant increase in access from the standard requirements; choose the user *or* proxy-user *accordingly. If you use a* user *or* proxy-user *with lesser permissions, no directory paths appear for the shares.*

For example, this shows all CIFS-share paths for the filer at 192.168.25.29. The "CELEBS$," "Y2004," "Y2005" shares are subshares of the "prescriptions" share:

```
bstnA# show exports host 192.168.25.29 paths proxy-user acoProxy2
Export probe of filer "192.168.25.29"

CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG
    Pre-Win2k Domain  MEDARCH

Paths:

  CIFS

    Share                            Directory
    ---------------------------      ------------------------------------
    CELEBS$                          e:\exports\prescriptions\VIP_wing
    Y2004                            e:\exports\prescriptions\2004
    Y2005                            e:\exports\prescriptions\2005
    prescriptions                    e:\exports\prescriptions
    ---------------------------      ------------------------------------
    Share                            Directory

bstnA# ...
```

Focusing on One Share

To focus on one share, use the share argument in the show exports command:

```
show exports host filer share share-name
[user username windows-domain domain | proxy-user proxy]
```

where

share-name (1-1024 characters) identifies the share, and

the other options are explained above.

This shows the default report, but only with entries that are relevant to the share. For example, the following command focuses on the 'histories' share.

```
bstnA> show exports host 192.168.25.20 share histories proxy-user acoProxy2
Export probe of filer "192.168.25.20"
```

```
Connectivity:

  Slot.Proc  Proxy Address    Ping (size: result)
  ---------  ---------------  -------------------
  2.1        192.168.25.31    64: Success  2000: Success  8820: Success
  2.2        192.168.25.32    64: Success  2000: Success  8820: Success
  2.3        192.168.25.33    64: Success  2000: Success  8820: Success
  2.4        192.168.25.34    64: Success  2000: Success  8820: Success
  2.5        192.168.25.141   64: Success  2000: Success  8820: Success
  2.6        192.168.25.142   64: Success  2000: Success  8820: Success
  2.7        192.168.25.143   64: Success  2000: Success  8820: Success
  2.8        192.168.25.144   64: Success  2000: Success  8820: Success
  2.9        192.168.25.145   64: Success  2000: Success  8820: Success
  2.10       192.168.25.146   64: Success  2000: Success  8820: Success
  2.11       192.168.25.147   64: Success  2000: Success  8820: Success
  2.12       192.168.25.148   64: Success  2000: Success  8820: Success

CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG
    Pre-Win2k Domain  MEDARCH

Capabilities:

  CIFS
    Server            TCP/445, TCP/139
    SMB:
      Security Mode   User level, Challenge/response, Signatures disabled
      Capabilities    Large Files, Large Read, Large Write, Info Passthru
      Max Buffer      16644
    Auth Protocols    NTLMSSP, Kerberos
    IPC$ Share        Access OK
    Auth Method Used  Kerberos
    SPN Used          vm-swp2003s2-04$@MEDARCH.ORG
    Discovered SPN    vm-swp2003s2-04$@MEDARCH.ORG

Shares:

  CIFS
                                                  Space
    Share                                   Total        Avail      Serial Num
    -------------------------------    ----------   ----------      ----------
    histories                              1.9 GB       1.4 GB       fe74-2087

Time:

  CIFS
    Filer's time is the same as the switch's time.
bstnA> ...
```

Showing Connectivity Only

Use the connectivity keyword to show the Connectivity table alone:

```
show exports host filer [share share-path] connectivity
```

where the options are explained above.

CIFS access is not required for this examination, so you do not need Windows-user credentials.

The ARX sends pings of various sizes from each proxy IP, to confirm that the Maximum Transmission Unit (MTU) is set consistently between each proxy IP and the filer. If any of these pings fails (but not all of them), the filer is unusable because some network equipment between the proxy IP and the filer has an inconsistent MTU setting. (To change the MTU on the ARX, refer to *Enabling Jumbo Frames (optional)*, on page 3-8 of the network guide.)

For example, this command shows good connectivity to 192.168.25.20:

```
bstnA> show exports host 192.168.25.20 connectivity
Export probe of filer "192.168.25.20"

Connectivity:

  Slot.Proc  Proxy Address    Ping (size: result)
  ---------  ---------------  -------------------
  2.1        192.168.25.31    64: Success  2000: Success  8820: Success
  2.2        192.168.25.32    64: Success  2000: Success  8820: Success
  2.3        192.168.25.33    64: Success  2000: Success  8820: Success
  2.4        192.168.25.34    64: Success  2000: Success  8820: Success
  2.5        192.168.25.141   64: Success  2000: Success  8820: Success
  2.6        192.168.25.142   64: Success  2000: Success  8820: Success
  2.7        192.168.25.143   64: Success  2000: Success  8820: Success
  2.8        192.168.25.144   64: Success  2000: Success  8820: Success
  2.9        192.168.25.145   64: Success  2000: Success  8820: Success
  2.10       192.168.25.146   64: Success  2000: Success  8820: Success
  2.11       192.168.25.147   64: Success  2000: Success  8820: Success
  2.12       192.168.25.148   64: Success  2000: Success  8820: Success
bstnA> ...
```

Showing Capabilities Only

The capabilities keyword shows only the Capabilities table:

```
show exports host filer [share share-path] capabilities [user
username windows-domain domain | proxy-user proxy]
```

where the options are explained earlier in the chapter.

For a CIFS filer, the Windows credentials are used to discover the filer's service-principal name (*SPN*), and to establish an authentication method (Kerberos, NTLMv2, or NTLM) used for the given ***username*** or ***proxy*** credentials. The ARX requires two configuration options to authenticate with Kerberos: the full FQDN for the Windows domain, and the filer's SPN.

For example, this filer supports NFS; the command does not provide any Windows credentials, so there is no Kerberos authentication.

```
bstnA> show exports host 192.168.25.19 capabilities
Export probe of filer "192.168.25.19"

Capabilities:
  NFS
    Port Mapper    TCP/111, UDP/111
    Mount Daemon   V1 TCP/959, V1 UDP/956, V2 TCP/959, V2 UDP/956, V3 TCP/959, V3 UDP/956
    Server         V2 UDP/2049, V3 TCP/2049, V3 UDP/2049
```

As another example, this command includes the proxy-user option to show the capabilities of a CIFS filer:

```
bstnA> show exports host 192.168.25.20 capabilities proxy-user acoProxy2
Export probe of filer "192.168.25.20"

CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG
    Pre-Win2k Domain  MEDARCH

Capabilities:

  CIFS
    Server            TCP/445, TCP/139
    SMB:
      Security Mode   User level, Challenge/response, Signatures disabled
      Capabilities    Large Files, Large Read, Large Write, Info Passthru
      Max Buffer      16644
    Auth Protocols    NTLMSSP, Kerberos
    IPC$ Share        Access OK
    Auth Method Used  Kerberos
    SPN Used          vm-swp2003s2-04$@MEDARCH.ORG
    Discovered SPN    vm-swp2003s2-04$@MEDARCH.ORG
```

Showing Shares Only

To list only the filer’s shares, use the shares keyword:

```
show exports host filer shares
[user username windows-domain domain | proxy-user proxy]
```

where the options are explained earlier.

The NFS table that appears if no user or proxy-user is specified shows each share path and the machines and/or subnets that can access the share. For a share to be eligible for import, all proxy-IP addresses must be on this list.

The CIFS table that appears if a user or proxy-user is specified shows two disk-space measures and the serial number for the storage volume behind the share. If two shares have the same serial number, they are assumed to be shares for the same storage volume on the filer.

For example:

```
bstnA> show exports host 192.168.25.20 shares proxy-user acoProxy2
Export probe of filer "192.168.25.20"

CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG
    Pre-Win2k         MEDARCH
```

```
Shares:

  CIFS
                                                 Space
    Share                                  Total        Avail      Serial Num
    ------------------------------    ----------   ----------     ----------

    share_e_0                             1.9 GB       1.5 GB      2881-232d
    share_e_1                             1.9 GB       1.5 GB      2881-232d
    share_e_10                            1.9 GB       1.5 GB      2881-232d
    share_e_11                            1.9 GB       1.5 GB      2881-232d
    share_e_12                            1.9 GB       1.5 GB      2881-232d
    share_e_13                            1.9 GB       1.5 GB      2881-232d
    share_e_14                            1.9 GB       1.5 GB      2881-232d
    share_e_15                            1.9 GB       1.5 GB      2881-232d
    share_e_16                            1.9 GB       1.5 GB      2881-232d
    share_e_17                            1.9 GB       1.5 GB      2881-232d
    share_e_18                            1.9 GB       1.5 GB      2881-232d
    share_e_19                            1.9 GB       1.5 GB      2881-232d
    ------------------------------    ----------   ----------     ----------
    Share                                  Total        Avail      Serial Num
                                                 Space
bstnA> ...
```

Showing CIFS Attributes

Each back-end CIFS share has several CIFS attributes that are relevant to namespace imports; if an ARX managed volume includes multiple shares, it can only support the attributes that are common to all of them. These attributes represent support for Access-Based Enumeration (ABE), Compressed Files, Named Streams, Persistent ACLs, Sparse Files, and/or Unicode file names on disk. This command shows a table of supported CIFS attributes at each of the filer's shares.

show exports host *filer* [share *share-path*] attributes
[user *username* windows-domain *domain* | proxy-user *proxy*]

where the options are explained earlier.

For example, this command shows the supported CIFS attributes for the "histories" share on filer 192.168.25.20.

```
bstnA> show exports host 192.168.25.20 share histories attributes proxy-user acoProxy2
Export probe of filer "192.168.25.20"

CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG
    Pre-Win2k Domain  MEDARCH

Attributes:

  CIFS
    Codes: AE=Access-based Enum, CF=Compressed Files, NS=Named Streams,
           PA=Persistent ACLs, SF=Sparse Files, UD=Unicode On Disk.
    Key:   X set, - not set, ? not reported (may need admin-level access)

                                          Attributes
    Share                           AE  CF  NS  PA  SF  UD
    ------------------------------  --  --  --  --  --  --
    histories                       -   X   X   X   X   X
bstnA> ...
```

Showing Time Settings

Namespace policy (described in later chapters) requires that the ARX has its clock synchronized with those of its back-end filers. Kerberos authentication also requires synchronized time. You should configure the ARX to use the same NTP servers that the filers use; refer to *Configuring NTP*, on page 4-15 in the *ARX® CLI Network-Management Guide* for instructions.

To check the clock skew between the ARX and a filer, use the time keyword in the show exports command:

show exports host *filer* [share *share-path*] [user *username* windows-domain *domain* | proxy-user *proxy*] time

where the options are explained earlier.

For example:

```
bstnA> show exports host 192.168.25.20 time proxy-user acoProxy2
Export probe of filer "192.168.25.20"

CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG
    Pre-Win2k         MEDARCH

Time:

  CIFS
    Filer's time is the same as the switch's time.
bstnA>
```

Probing for CIFS Security

You can run a probe test to verify that your Windows credentials are adequate for one or more back-end shares. This test applies to CIFS shares only.

This examination attempts to write to the filer's shares and determines whether or not the ***username*** or ***proxy-user*** has (at least) Backup Operator privileges at each share. Both Write and Privs should show "OK" status for the share to be eligible for import. If either test fails, try another proxy user, or find a ***username*** that works and add a new proxy user with those credentials (as shown in *Adding a Proxy User*, on page 3-4).

This filer examination is more intrusive than the others, so it is not invoked as part of show exports.

From priv-exec mode, use the probe exports command to test some Windows credentials at a given back-end filer:

```
probe exports host filer [spn spn] [share share-path]
{user username windows-domain domain | proxy-user proxy-user}
```

where the options match those of the show exports command, above. This includes the information that is returned if a user or proxy-user is specified (CIFS only) or not (NFS only).

For example, this command tests the "acoProxy2" credentials at a single back-end share ('histories' on filer 192.168.25.20):

```
bstnA> enable
bstnA# probe exports host 192.168.25.20 share histories proxy-user acoProxy2
Export probe of filer "192.168.25.20"

CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG
    Pre-Win2k         MEDARCH

Security:

  CIFS

    Description Key: OK (success) NO (failure) -- (not applicable)

    Share                            Write   Privs
    -------------------------------  -----   -----
    histories                        OK      OK
bstnA# ...
```

6

Adding an External Filer

Concepts and Terminology

An ARX *volume* stores its files on your back-end filers and servers. A *managed volume* keeps track of all file and directory locations in an external database, called the volume's *metadata*. A managed volume builds this database of metadata when you first enable it; it scans all of its back-end servers and *imports* all files and directories by recording their locations. By keeping metadata, a managed volume can support policies for balancing or migrating files.

You can use your filers and servers to store client files, metadata, or both.

Overview

A Network-Attached Storage (NAS) filer or a file server with Direct-Attached Storage (DAS) is configured as an *external filer* on the ARX. An external filer defines how to access the storage in an external NAS/DAS-based device. From gbl mode, use the external-filer command to create an empty external-filer instance:

```
external-filer name
```

where ***name*** (1-64 characters) is a name you choose for this external filer. This is often associated with the machine's NetBIOS name or hostname, but you can choose any name.

The CLI prompts for confirmation before creating the external-filer object in the database. Enter **yes** to confirm. (You can disable all such create-confirmation prompts with the terminal expert command.)

This puts you into gbl-filer mode. From gbl-filer mode, you need to identify the IP address(es) of the external filer. You can also provide a description and other optional parameters.

For example, the following command sequence creates an external filer named "das1:"

```
bstnA(gbl)# external-filer das1
This will create a new filer.

Create filer 'das1'? [yes/no] yes
bstnA(gbl-filer[das1])# ...
```

Providing the Filer's IP Address

The next step in external-filer configuration is to give the IP address of the filer. The address must be on the proxy-IP subnet (*Adding a Range of Proxy-IP Addresses*, on page 4-7 of the *ARX® CLI Network-Management Guide*) or reachable through a gateway on that subnet (via static route: see *Adding a Static Route*, on page 4-10 of the same manual).

Use the ip address command to identify the external filer:

```
ip address ip-address
```

where ***ip-address*** is in dotted-decimal format (for example, 192.168.25.19).

For example, the following command sequence declares the external filer, "das1," to be the NAS filer at 192.168.25.19:

```
bstnA(gbl)# external-filer das1
bstnA(gbl-filer[das1])# ip address 192.168.25.19
bstnA(gbl-filer[das1])# ...
```

Providing a Secondary Address (UDP only)

Some filers are configured as redundant pairs that share a single virtual-IP address but occasionally respond to UDP packets from their physical-IP addresses. That is, a client (the ARX) sends a request to the virtual-IP address, and one of the filers responds from its physical-IP address. The switch considers the virtual IP to be *primary*, and the physical IPs to be *secondary*. To identify a secondary IP address, use the secondary flag in the ip address command:

```
ip address ip-address secondary
```

where ***ip-address*** is a secondary address for the filer.

This only applies to a filer that you will access through UDP; that is, with NFSv2 or NFSv3/UDP. If you only plan to use CIFS and/or NFSv3/TCP, the secondary address is unnecessary.

You can use this flag to configure up to four secondary IP addresses. For example, the following command sequence configures an external filer, "nas1," with one primary address and two secondary addresses:

```
bstnA(gbl)# external-filer nas1
bstnA(gbl-filer[nas1])# ip address 192.168.25.21
bstnA(gbl-filer[nas1])# ip address 192.168.25.61 secondary
bstnA(gbl-filer[nas1])# ip address 192.168.25.62 secondary
bstnA(gbl-filer[nas1])# ...
```

Removing a Secondary Address

Use the no form of the command to remove a secondary IP address from the list:

```
no ip address ip-address secondary
```

where ***ip-address*** is the secondary address to remove.

For example:

```
bstnA(gbl)# external-filer nas1
bstnA(gbl-filer[nas1])# no ip address 192.168.25.65 secondary
bstnA(gbl-filer[nas1])# ...
```

Changing the IP Address

It is possible to change the IP address of one or more backend filers without having to re-import the filesystems. Use the ip address change-to command in gbl-filer mode to change an external filer’s current IP address in the ARX configuration, then use the ext-filer-ip-addrs activate command in priv-exec mode to reboot the ARX (and its redundant peer, if there is one) and cause the change to take effect. A good time to execute the IP address change on the filer itself is between the time you change that IP address in the ARX configuration and before you activate the configuration on the ARX.

For example:

Change the external filer's IP address using this command:

```
ip address ip-addr change-to new-ip-addr
```

where *ip-addr* is the filer’s current IP address; and

new-ip-addr is the address to which you want to changer the filer.

The system responds with a reminder that this change will not take effect until after a failover or a configuration reload, and prompts you to confirm that you want to change the IP address.

Make the change take effect using this command:

```
ext-filer-ip-addrs activate
```

This reboots the ARX and its redundant peer; once the reboot is complete, all external filer IP address changes configured using ip address... change-to take effect. The reboot causes a service outage, so you should make sure to execute the command only during non-busy hours. The CLI prompts you for confirmation before rebooting; type yes to proceed.

Use the show global-config command to display the new IP address prior to the execution of failover.

The ip address... change-to command has a “no” form, as well, which removes the new IP address. For example:

```
no ip address ip-addr change-to new-ip-addr
```

If the new IP address does not exist in the temporary database already, attempting to remove it produces an error message.

If a copy of the global config is saved prior to the failover, and is re-played after the failover, this command will result in an error message, because the primary IP address will be incorrect.

Ignoring a Snapshot or Checkpoint Directory

Some filer shares include read-only directories that should not be aggregated into a namespace, such as snapshot and checkpoint directories. The .snapshot directory, used for backups on some NAS devices, is ignored by default. To exclude another directory (or any file), use the ignore-name command from gbl-filer mode:

```
ignore-name directory
```

where ***directory*** (1-256 characters) is the directory to ignore. Do not use any slash (/) characters in this directory name (for example, "dir" is valid, but "dir/subdir" is not).

If the directory name starts with a "." and is at least three characters long, you can use an asterisk (*) at the end as a wildcard. These wildcards only apply to the root of the share: ".ckpt*" ignores /.ckpt2, but does not ignore /docs/.ckpt7.

The CLI prompts for confirmation if you use a wildcard. Enter **yes** to continue.

You can ignore up to eight directories per external filer. These are common directories to ignore for various filer vendors:

- EMC: .etc, lost+found, .ckpt*
- EMC Data Domain: .snapshot
- NetApp (formerly Network Appliance): .snapshot, ~snapshot
- Windows: "System Volume Information"
- Hitachi HNAS (BlueArc): .snapshot, ~snapshot

◆ **Note**

Ignore only special, virtual directories designed for filer backups, or directories that only appear in the share's root. If you ignore a standard directory below the root, a client cannot delete the directory's parent.

For example, the following command sequence ignores several directories in the external filer, "nasE1:"

```
bstnA(gbl)# external-filer nasE1
bstnA(gbl-filer[nasE1])# ignore-name .etc
bstnA(gbl-filer[nasE1])# ignore-name lost+found
bstnA(gbl-filer[nasE1])# ignore-name .ckpt*
Wildcard matches are only made in the root directory of a share.
Is this the intended behavior? [yes/no] yes
bstnA(gbl-filer[nasE1])# ...
```

Re-Instating a Directory

For cases where you want to start using a previously-ignored directory (and/or file), use the no ignore-name command from gbl-filer mode:

```
no ignore-name directory
```

where ***directory*** (1-256 characters) is the directory to stop ignoring.

For example, the following command sequence re-instates the ".back" directory for the external filer, "das1:"

```
bstnA(gbl)# external-filer das1
bstnA(gbl-filer[das1])# no ignore-name .back
bstnA(gbl-filer[das1])# ...
```

Adding a Description (optional)

You can add a description to the filer for use in show commands. The description can differentiate the external filer from others. From gbl-filer mode, use the description command to add a description:

```
description text
```

where ***text*** is 1-255 characters. Quote the text if it contains any spaces.

For example:

```
bstnA(gbl)# external-filer das1
bstnA(gbl-filer[das1])# description "shares with financial data (LINUX filer, rack 14)"
bstnA(gbl-filer[das1])# ...
```

Removing the Description

From gbl-filer mode, use no description to remove the description string from the current external-filer configuration:

```
no description
```

For example:

```
bstnA(gbl)# external-filer fs22
bstnA(gbl-filer[fs22])# no description
bstnA(gbl-filer[fs22])# ...
```

Setting the CIFS Port (optional)

By default, the ARX sends its CIFS messages to port 445 or 139 at the external filer. Port 445 supports raw CIFS communication over TCP, port 139 supports CIFS through NetBIOS; the ARX tries port 445 first and uses port 139 as a fall-back. For most (if not all) CIFS configurations, this should suffice. For filers that do not listen at either of these well-known ports, use the cifs-port command to set the port:

```
cifs-port port
```

where ***port*** is a TCP/UDP port number from 1-65535. Numbers from 0-1023 are reserved for "well-known" TCP/UDP ports.

For example, the following command sequence sets the CIFS port to 7231 for the filer named "fs1:"

```
bstnA(gbl)# external-filer fs1
bstnA(gbl-filer[fs1])# cifs-port 7231
bstnA(gbl-filer[fs1])# ...
```

Refer to RFCs 1001 and 1002 for NetBIOS specifications.

Source Computer Name Used for NetBIOS (139) Connections

Whenever the ARX falls back to port 139 to make a NetBIOS connection, it uses a "Source Computer" name of "ACOPIA_SWITCH." Every proxy-IP address identifies itself by this same name. You can see this name through MMC, by examining "Active Sessions" on the back-end filer.

Reverting to the CIFS-Port Default

Use no cifs-port to revert back to the default CIFS port, 445 (or 139):

```
no cifs-port
```

For example, the following command sequence stops using a pre-set CIFS port for the filer named "fs1:"

```
bstnA(gbl)# external-filer fs1
bstnA(gbl-filer[fs1])# no cifs-port
bstnA(gbl-filer[fs1])# ...
```

Setting the SPN (CIFS)

This section only applies to a filer that supports CIFS and

- is a Windows cluster, or
- supports Kerberos authentications.

This is especially important for a Windows 2008 cluster, which requires this command to function behind the ARX. You can skip to the next section if this is an NFS-only filer, or if it is not a Windows cluster *and* it does not support Kerberos.

The ARX software uses the filer's *service principal name* (SPN) for all Kerberos authentications. On Windows 2008 clusters, the virtual SPN is required to even connect to the CIFS service. For Windows 2003 clusters, the virtual SPN is required for the ARX to retain its connection(s) after a cluster failover.

◆ **Note**

A virtual SPN is therefore required on any Windows cluster behind the ARX. Configure a SPN that the ARX can use before and after any cluster failover.

For filers other than Windows clusters, the ARX software may be able to automatically discover the SPN. The SPN-discovery process occurs when a managed volume imports its first share from a filer, or when you use the show exports command to show a filer's CIFS capabilities (recall *Showing Capabilities Only*, on page 5-11).

To manually set the SPN, use the gbl-filer spn command:

```
spn hostname@domain
```

where the ***hostname@domain*** (1-256 characters total) is the SPN. Do not include any "$" or "/" characters in this string. The SPN should contain these three items, in order:

hostname is the filer's hostname, or the virtual service name shared by the Windows cluster.

@ is a required character. There must be exactly one "@" character between the filer's hostname and the domain.

domain is the Windows domain for the filer. This must be a full FQDN (for example, "mydiv.myco.com" instead of just "mydiv").

Once you set this SPN, the ARX ignores the discovered SPN (if there is one).

For example, this command sequence sets a configured SPN for the external filer named "fs4:"

```
bstnA(gbl)# external-filer fs4
bstnA(gbl-filer[fs4])# spn fs4@medarch.org
bstnA(gbl-filer[fs4])# ...
```

Note

If the back-end filer changes its hostname, its SPN changes, also. Invoke this command again to establish the new SPN after the filer name changes. Until the SPN is updated, no managed volume can perform autonomous operations on the filer, such as importing files and migrating them to or from one of the filer's shares, and CIFS clients have limited access through Kerberos.

Note

For information about using an SPN in conjunction with constrained delegation, refer to Using an SPN With Constrained Delegation in a Two-Tier Configuration, on page 11-31.

Limiting CIFS Connections to the Filer

Some Tier-2 filers can only tolerate a limited number of simultaneous CIFS connections. You have the option to limit the CIFS connections from the ARX to any such back-end filer. From gbl-filer mode, use the cifs connection-limit command to impose a limit on CIFS connections to the current filer:

```
cifs connection-limit maximum
```

where ***maximum*** (at least 250 in an ARX-4000, at least 150 in the smaller chassis models) is the maximum number of CIFS connections to the current back-end filer.

If the filer is being used and this limit is lower than the current CIFS connections to it, the CLI prompts you with a choice: the ARX can wait for clients to disconnect gracefully, or break all connections immediately. If you choose to wait and the client connections persist, you can later use another CLI command to forcibly break all connections to the filer. For instructions on breaking the connections later, refer to *Dropping All Connections to a Filer*, on page 9-30 of the *ARX CLI Maintenance Guide*.

For example, the following command sequence sets a limit of 500 connections to the external filer, "smb1:"

```
bstnA(gbl)# external-filer smb1
bstnA(gbl-filer[smb1])# cifs connection-limit 500
bstnA(gbl-filer[smb1])# ...
```

Removing the Connection Limit

To remove the connection limit, use the no form of the cifs connection-limit command:

```
no cifs connection-limit
```

For example, the following command sequence allows for unlimited CIFS connections to the NAS10 filer:

```
bstnA(gbl)# external-filer nas10
bstnA(gbl-filer[nas10])# no cifs connection-limit
bstnA(gbl-filer[nas10])# ...
```

Preparing the Filer for ARX-Snapshot Support

The ARX can coordinate snapshots and/or checkpoints at multiple filers and present them to its clients as a single, consistent snapshot. To coordinate the snapshots, the ARX logs into the filers' command lines, issues snapshot or checkpoint commands at each filer, and waits for the results of each operation.

If your external filer can support snapshots or checkpoints and you plan to use it this way, you must supply information for accessing its CLI and issuing commands. Otherwise, you can skip this section.

Before You Begin

A limited number of vendors and software releases have been qualified for ARX-snapshot support. Consult the ARX Release Notes for your software release to confirm that your filer or file server is supported. For supported filers, you must create an additional proxy user for accessing the filer's CLI.

Creating a Proxy User for Accessing the Filer's CLI

To invoke snapshots at the back-end filer, the ARX requires administrative privileges at the filer's CLI. You add the proper administrative username and password as a proxy user, which typically holds Windows-client credentials (recall *Adding a Proxy User*, on page 3-4).

For EMC and NetApp equipment, these are UNIX credentials that do not require a Windows domain. They are for SSH or RSH logins only. For Windows servers, these credentials do require the server's Windows Domain.

For example, this command sequence adds a proxy user named "nas_admin" for logging into NetApp or EMC devices:

```
bstnA(gbl)# proxy-user nas_admin
bstnA(gbl-proxy-user[nas_admin])# user root
Password: rootpasswd
Validate Password: rootpasswd
bstnA(gbl-proxy-user[nas_admin])# exit
bstnA(gbl)# ...
```

As another example, this command sequence adds a proxy user named "cifs_admin" for logging into Windows servers:

```
bstnA(gbl)# proxy-user cifs_admin
bstnA(gbl-proxy-user[cifs_admin])# user Administrator
Password: adminpasswd
Validate Password: adminpasswd
bstnA(gbl-proxy-user[cifs_admin])# windows-domain MEDARCH.ORG
bstnA(gbl-proxy-user[cifs_admin])# exit
bstnA(gbl)# ...
```

Identifying the Filer Type

The vendor of the filer is required to determine the correct set of CLI commands for invoking snapshots or checkpoints. From gbl-filer mode, use the filer-type command to identify the filer vendor:

```
filer-type {bluearc | emc | data-domain | network-appliance | windows}
```

For example, the following command sequence indicates that the filer named “nas10” is a NetApp (formerly Network-Appliance) filer:

```
bstnA(gbl)# external-filer nas10
bstnA(gbl-filer[nas10])# filer-type network-appliance
bstnA(gbl-filer[nas10])# ...
```

Choosing the Management Protocol (NetApp Only)

By default, the ARX uses the Secure SHell (SSH) to access the filer’s CLI. Some NetApp filers support only the Remote SHell (RSH). For those filers, add management-protocol rsh to the end of the filer-type command:

```
filer-type network-appliance management-protocol rsh
```

For example, the following command sequence indicates that the ARX should use RSH to access the “nas10” filer:

```
bstnA(gbl)# external-filer nas10
bstnA(gbl-filer[nas10])# filer-type network-appliance management-protocol rsh
bstnA(gbl-filer[nas10])# ...
```

Reverting to the Default

Rerun the filer-type command with management-protocol ssh to revert to SSH:

```
filer-type network-appliance management-protocol ssh
```

For example, the following command sequence selects SSH access for the “nas10” filer:

```
bstnA(gbl)# external-filer nas10
bstnA(gbl-filer[nas10])# filer-type network-appliance management-protocol ssh
bstnA(gbl-filer[nas10])# ...
```

Choosing the NAS_DB Path (EMC Celerra Only)

EMC Celerra servers have a Unix “path” variable, NAS_DB, that is required for performing checkpoints on the server. By default, the ARX uses “/nas” as the value for the NAS_DB variable. This is also the default for an EMC Celerra server. Some sites use EMC’s Symmetrix Remote Data Facility (SRDF) to synchronize the data on two file servers; in this configuration, the destination file server uses a different NAS_DB path than the source. (Contact EMC Support if you have questions about the proper setting for this variable on your EMC server.)

For an EMC file server with a different path, use the emc nas-db-path argument at the end of the filer-type command:

```
filer-type emc nas-db-path path
```

where ***path*** (1-64 characters) is the value for the NAS database (NAS_DB) path variable.

For example, the following command sequence indicates that the ARX should use “/nas/rdf/500” as the NAS_DB setting on the “nasE2” server:

```
bstnA(gbl)# external-filer nasE2
bstnA(gbl-filer[nasE2])# filer-type emc nas-db-path /nas/rdf/500
bstnA(gbl-filer[nasE2])# ...
```

Reverting to the Default

The default NAS_DB value is “/nas.” Use the filer-type emc command. without the nas-db-path option, to revert to this default:

```
filer-type emc
```

For example, the following command sequence resets the NAS_DB path for the “nasE1” filer:

```
bstnA(gbl)# external-filer nasE1
bstnA(gbl-filer[nasE1])# filer-type emc
bstnA(gbl-filer[nasE1])# ...
```

Preparing a Windows Server with WinRM

The ARX invokes snapshots on a Windows server through Windows Remote Management (WinRM). WinRM is the Microsoft implementation of a standard, SOAP-based management protocol. To use ARX snapshots on a Windows server, the server requires an installation of WinRM and a running WinRM listener. The minimum configuration is a WinRM listener that allows HTTP negotiations (encrypted or unencrypted), and allows up to five minutes before timing out a snapshot operation. The ARX uses Kerberos authentication over HTTP.

To implement any such Kerberos authentications, the ARX must discover the domain controller (DC) for the proxy user’s domain. This requires knowledge of the local Active-Directory (AD) forest and its Windows domains. For instructions on automatically discovering the AD forest, recall *Discovering the Active-Directory Forest (Kerberos)*, on page 3-9.

By default, the ARX sends its WinRM directives to port 80. If your WinRM listener is listening on a different port, you can use the port option to specify the correct port number:

```
filer-type windows port number
```

where ***number*** (1-65535) is the port number where the WinRM listener is waiting for management instructions.

For example, the following command sequence specifies that the “fs4” server is waiting for WinRM commands at port 5985:

```
bstnA(gbl)# external-filer fs4
bstnA(gbl-filer[fs4])# filer-type windows port 5985
bstnA(gbl-filer[fs4])# ...
```

You can use the following command (at the Windows filer's DOS prompt) to create a five-minute timeout for snapshots:

winrm set winrm/config@{MaxTimeoutms="300000"}.

Reverting to the Default WinRM Port

The default port for WinRM commands is "80," the port reserved for HTTP transmissions. Use the filer-type windows command. without the port option, to revert to this default:

```
filer-type windows
```

For example, the following command sequence resets the WinRM for the "fs4" filer:

```
bstnA(gbl)# external-filer fs4
bstnA(gbl-filer[fs4])# filer-type windows
bstnA(gbl-filer[fs4])# ...
```

Removing the Filer Type

Use the no filer-type command to remove the filer-type designation from this external filer. This indicates that the filer cannot support ARX snapshots:

```
no filer-type
```

This is the default.

For example:

```
bstnA(gbl)# external-filer nas2
bstnA(gbl-filer[nas2])# no filer-type
bstnA(gbl-filer[nas2])# ...
```

Selecting a Separate IP for Filer Management

By default, the ARX accesses the filer's CLI through its primary-IP address (recall *Providing the Filer's IP Address*, on page 6-4). Some file servers use a separate, out-of-band interface for accessing the CLI. For these file servers, you must specify the management address. To enter a separate management address, use the management flag at the end of the ip address command:

```
ip address ip-address management
```

where ***ip-address*** is a separate management address for the filer.

For example, the following command sequence configures an external filer, "nasE1," with a separate, out-of-band management address:

```
bstnA(gbl)# external-filer nasE1
bstnA(gbl-filer[nasE1])# ip address 192.168.25.52 management
bstnA(gbl-filer[nasE1])# ...
```

Reverting to the Primary IP for Management

You can use the no form of the command (with the management flag) to remove the separate management IP. This causes the ARX to access the CLI through the filer's primary IP:

```
no ip address ip-address management
```

where ***ip-address*** is the separate management address that was previously assigned to the filer.

For example, the following command sequence removes the out-of-band-management address from the "nas6" filer:

```
bstnA(gbl)# external-filer nas6
bstnA(gbl-filer[nas6])# no ip address 192.168.25.93 management
bstnA(gbl-filer[nas6])# ...
```

Supplying Credentials for Management Access

The ARX requires a username and password to access the filer's management functions. The ARX logs into a NetApp or EMC device's CLI through SSH or RSH; for a Windows server, the ARX uses Windows Remote Management (WinRM). In either case, the username and password should be in a pre-configured proxy user, as described earlier (recall *Creating a Proxy User for Accessing the Filer's CLI*, on page 6-12). This proxy user does not require a Windows domain for NetApp or EMC access, but does require a fully-qualified domain (such as "mydiv.myco.com") for accessing a Windows server.

Use the gbl-filer proxy-user command to apply the proxy user to the current external filer:

```
proxy-user name
```

where ***name*** (1-32 characters) identifies the proxy user for this filer's CLI. Use the show proxy-user command for a list of configured proxy users.

For example, this command set applies a proxy user, "nas_admin," to the "nas10" filer:

```
bstnA(gbl)# external-filer nas10
bstnA(gbl-filer[nas10])# proxy-user nas_admin
bstnA(gbl-filer[nas10])# ...
```

Allowing the ARX to Manage the Filer's Snapshots

The final step in permitting ARX snapshots is to explicitly confirm that they are allowed. Use the manage snapshots command to confirm that the ARX should manage snapshots at the current filer:

```
manage snapshots
```

For example, the following command sequence permits ARX snapshots on the "nas10" filer:

```
bstnA(gbl)# external-filer nas10
bstnA(gbl-filer[nas10])# manage snapshots
bstnA(gbl-filer[nas10])# ...
```

Preventing ARX Snapshots at a Filer

If a snapshot-capable filer begins to run out of space or has some other issue that disqualifies it from keeping snapshots, you can use the no manage snapshots command:

```
no manage snapshots
```

This prevents the ARX from invoking snapshots on any of the filer's shares. If snapshots are allowed on the filer later, you can then use the affirmative form of the same command (described above) to permit the ARX to invoke snapshots.

For example, the following command sequence prevents new ARX snapshots on the “nas6” filer:

```
bstnA(gbl)# external-filer nas6
bstnA(gbl-filer[nas6])# no manage snapshots
bstnA(gbl-filer[nas6])# ...
```

Listing External Filers

Use the show external-filer command to get a list of all external filers:

show external-filer

For example, the following command lists all of the external filers known to the ARX:

```
bstnA# show external-filer
  Name                        IP Address     Description
  ------------------------    -------------  ----------------------------
  das1                        192.168.25.19  financial data (LINUX filer, rack 14)
  fs1                         192.168.25.20  misc patient records (Table 3)
  fs2                         192.168.25.27  bulk storage server (Table 3)
  fs3                         192.168.25.28  Hematology lab server (Table 8)
  fs4                         192.168.25.29  prescription records (Table 3)
  fs5                         192.168.25.71  docs, invoices, for scanners (Table 7)
  fs6                         192.168.25.30  records of lab animals - lab rats (rack C)
  fs7                         192.168.25.41  lab animal test results - lab rats (rack C)
  das2                        192.168.25.22  Solaris filer 2 (rack 16)
  das3                        192.168.25.23  Solaris filer 3 (rack 16)
  nas1                        192.168.25.21  NAS filer 1 (rack 31)
                              192.168.25.61   (1st secondary)
                              192.168.25.62   (2nd secondary)
  das7                        192.168.25.24  Redhat-LINUX filer 1
  das8                        192.168.25.25  Redhat-LINUX filer 2
  nas2                        192.168.25.44  NAS filer 2 (rack 31)
  nas3                        192.168.25.47  NAS filer 3 (rack 32)
  nas10                       192.168.25.49  NAS filer 10 (rack 38)
  nas11                       192.168.25.48  filer 11 (rack 38)
  nasE1                       192.168.25.51  NAS filer E1
  smb1                        192.168.25.48  Samba filer
bstnA#
```

Showing External-Filer Details

Identify a particular filer with the show external-filer command to see details about the filer:

show external-filer *filer-name*

where ***filer-name*** (up to 64 characters) is the name given to the external filer.

For example, the following command shows the DAS filer named “das1:”

```
bstnA# show external-filer das1

Filer "das1" Configuration

  Description:                               financial data (LINUX filer, rack 14)
  Filer IP:                                  192.168.25.19
  Discovered SPN:                            Not discovered
  Filer Management Proxy User:
  Filer Management IP:                       192.168.25.19
  Filer Manage Snapshots:                    No
  CIFS Port:                                 auto-detect
  CIFS Connection Limit Overall:             none
  CIFS Connection Limit / NSM processor:     none
  NFS TCP Connections:                       1
```

```
Managed Exports
--------------------------------------------------------------------------------

  NFS Export: /exports/budget
    Namespace: wwmed
    Volume: /acct


  Directories and/or files to ignore for importing
  ------------------------------------------------
    .snapshot
    ~snapshot
    .etc
    .ckpt*
    $RECYCLE.BIN
    System Volume Information
    SIS Common Store
    .clusterConfig

bstnA# ...
```

As another example, the following command shows an external filer that can support ARX snapshots and SMB2:

```
bstnA# show external-filer nas10

Filer "nas10" Configuration

  Description:                               NAS filer 10 (rack 38)
  Filer IP:                                  192.168.25.49
  Discovered SPN:                            enterprise$@MEDARCH.ORG
  Filer Management Proxy User:               nas_admin
  Filer Management IP:                       192.168.25.49
  Filer Manage Snapshots:                    Yes
  CIFS Port:                                 445
  CIFS Connection Limit Overall:             none
  CIFS Connection Limit / NSM processor:     none
  NFS TCP Connections:                       1
  Filer Type:                                NetApp
   Management Protocol:                        RSH

CIFS Filer Capabilities
-----------------------
  SMB: Max Buffer 33028
       Large Files, Large Read, Large Write
  SMB2.002: Max Read 65536, Max Write 65536, Max Transact 65536

Managed Exports
--------------------------------------------------------------------------------

  CIFS Share: equipment
    Namespace: medarcv
    Volume: /lab_equipment

  CIFS Share: for_lease
    Namespace: medarcv
    Volume: /lab_equipment


  Directories and/or files to ignore for importing
  ------------------------------------------------
    .snapshot
    ~snapshot
    .etc
    .ckpt*
    $RECYCLE.BIN
    System Volume Information
    SIS Common Store
    .clusterConfig

bstnA# ...
```

Showing Details for all External Filers

Use show external-filer all to see details about every configured external filer:

```
show external-filer all
```

Samples - Adding Two Filers

The following command sequence adds two filers to an ARX:

```
bstnA(gbl)# external-filer das1
This will create a new filer.

Create filer 'das1'? [yes/no] yes
bstnA(gbl-filer[das1])# ip address 192.168.25.19
bstnA(gbl-filer[das1])# exit
bstnA(gbl)# external-filer fs1
This will create a new filer.

Create filer 'fs1'? [yes/no] yes
bstnA(gbl-filer[fs1])# ip address 192.168.25.20
bstnA(gbl-filer[fs1])# exit
bstnA(gbl)#
```

Removing an External Filer

You can remove an external filer from back-end storage by deleting its configuration. To be eligible for deletion, the external filer must not be referenced by any namespace. Use the show external-filer command (see *Showing External-Filer Details*, on page 6-18) to see if a namespace is referencing the filer.

Use the no form of the external-filer command to remove an external-filer instance:

no external-filer *nas-name*

where ***nas-name*** (1-64 characters) identifies the external filer.

For example, the following command sequence removes an external filer named “finances:”

```
bstnA(gbl)# no external-filer finances
bstnA(gbl)# ...
```

Next

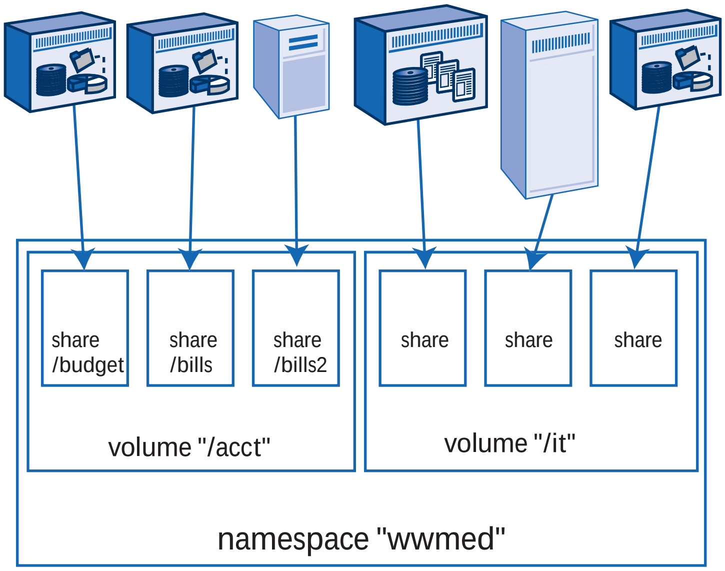
The external (NAS, or DAS-based) filer’s shares are ready to be included in a namespace volume. The next chapters describe how to configure namespaces and various types of volumes.

7

Configuring a Namespace

Overview

The ARX aggregates storage from external servers into one or more *namespaces*. A namespace is a collection of virtual file systems, called *volumes*. Each volume consists of storage space from any number of Network-Attached Storage (NAS) or filer servers with Direct-Attached Storage (DAS). A volume can contain multiple *shares*, where each share maps to an export or share from an external (NAS/DAS) filer. Clients access the volume as a single directory structure through a single mount point; the volume is analogous to a UNIX partition or a Windows logical drive.



The purpose of the namespace is to contain one or more volumes with a common set of access protocols (CIFS and/or NFS), authentication mechanisms, and character encoding. This chapter explains how to create a namespace. The next chapters explain how to aggregate your storage into various types of namespace volumes.

From gbl mode, use the namespace command to create a new namespace.

namespace *name*

> where ***name*** (1-30 characters) is a name you choose for the namespace. This is only a configuration name. It is not visible to clients.

The CLI prompts for confirmation before creating the namespace; this provides an opportunity to check for mistakes in the name. Enter **yes** to proceed. This puts you into gbl-ns mode, where you configure the volumes for the namespace.

For example, this command set creates a namespace called “wwmed:”

```
bstnA(gbl)# namespace wwmed
This will create a new namespace.

Create namespace 'wwmed'? [yes/no] yes
bstnA(gbl-ns[wwmed])# ...
```

Concepts and Terminology

A namespace can contain up to three types of volume: managed, direct, or shadow.

A *managed* volume keeps track of all files and directories in the servers behind it. The file and directory locations are called the volume’s *metadata*. A managed volume builds its database of metadata when you first enable it; it scans all of its back-end servers and *imports* all files and directories by recording their locations. By keeping metadata, a managed volume can support policies for balancing or migrating files.

A *direct* volume is a collection of mount points into its back-end storage. It does not import files and directories, does not keep metadata, and does not support policies. The direct volume is useful for quickly aggregating storage into a single mount point.

The *shadow* volume is a frequently-updated duplicate of a managed volume, possibly hosted at a different ARX in the same Resilient-Overlay Network (RON, described in the *ARX® CLI Network-Management Guide*).

Listing All Namespaces

To verify a new namespace was created, use the show namespace command to get a list of all namespaces:

```
show namespace
```

For example, the following command lists all of the namespaces known to the ARX:

```
bstnA# show namespace

Configured Namespaces
---------------------

Namespace                       Description
------------------------------  ----------------------------------------------
medco
wwmed                           namespace for World-Wide Medical network
medarcv
insur                           WW Medical insurance claims and records
labResearch
bstnA# ...
```

Showing Namespace Details

As you configure the namespace, it is useful to periodically view the namespace’s current configuration and running state. By including a namespace name, you can use the show namespace command to see details about the namespace and its volumes.

```
show namespace name
```

where ***name*** (1-30 characters) is the name of the namespace.

In addition to showing the full configuration of the namespace (including a number of components described later in this chapter), this report shows whether or not each namespace share is online.

For example, the following command shows the configuration and status of the namespace named “wwmed.”

```
bstnA# show namespace wwmed

Namespace "wwmed" Configuration
Description: namespace for World-Wide Medical network
Metadata Cache Size: 512 MB
NFS Character Encoding: ISO-8859-1

Supported Protocols
-------------------
  nfsv3 nfsv3-tcp

Participating Switches
----------------------
  bstnA (Volume Group 1) [Current Switch]


Volumes
-------
  /acct

             Volume freespace: 252 GB (automatic)
           Volume total space: 296 GB
                Metadata size: 1.9 MB
          Metadata free space: 62 GB
                    Snapshots: Not Enabled
             Migration method: Staged
                        State: Enabled

                  Host Switch: bstnA
                     Instance: 2
                 Volume Group: 1
                    Processor: 1.1
                        Files: 4,433 used (439 dirs), 3.9 M free, 252 M max (automatic)
    Metadata shares:

      Filer          Backend Path           Contains Metadata  Status
      -------------------------------------------------------------------
      nas1           /vol/vol1/meta1        Yes                Online

    Share bills
      Filer                        das8 [192.168.25.25]
      NFS Export                   /work1/accting
      Features                     unix-perm
      Status                       Online
      Critical Share               Yes
      Import Sync Attributes       Yes
      Import Priority              65535 (Lowest)
      Free space on storage        36 GB (38,783,361,024 B)
      Total space on storage       70 GB (75,278,499,840 B)
      Policy Maintain Freespace:   2 %
      Policy Resume Freespace:     3 %
      Free files on storage        17M
      Transitions                  1
      Last Transition             Wed 26 Jan 2011 01:05:14 AM EST

    Share bills2
      Filer                       das3 [192.168.25.23]
      NFS Export                  /exports/acct2
      Features                    unix-perm
      Status                      Online
      Import Sync Attributes      Yes
      Import Priority             65535 (Lowest)
      Free space on storage       111 GB (119,915,668,480 B)
      Total space on storage      113 GB (121,333,164,032 B)
      Policy Maintain Freespace:  2 %
      Policy Resume Freespace:    3 %
      Free files on storage       13M
      Transitions                 1
      Last Transition             Wed 26 Jan 2011 01:05:15 AM EST

    Share budget
      Filer                       das1 [192.168.25.19]
      NFS Export                  /exports/budget
      Features                    unix-perm
      Status                      Online
      Volume Root Backing         Yes
      Import Priority             65535 (Lowest)
      Free space on storage       81 GB (87,939,133,440 B)
      Total space on storage      89 GB (95,607,627,776 B)
      Policy Maintain Freespace:  2 %
      Policy Resume Freespace:    3 %
      Free files on storage       11M
      Transitions                 1
      Last Transition             Wed 26 Jan 2011 01:05:07 AM EST

    Share it5
      Filer                       das7 [192.168.25.24]
      NFS Export                  /lhome/it5
      Features                    unix-perm
      Status                      Online
      Import Sync Attributes      Yes
      Import Priority             65535 (Lowest)
      Free space on storage       22 GB (24,100,302,848 B)
      Total space on storage      24 GB (25,859,268,608 B)
      Policy Maintain Freespace:  2 %
      Policy Resume Freespace:    3 %
      Free files on storage       12M
      Transitions                 1
      Last Transition             Wed 26 Jan 2011 01:05:17 AM EST

bstnA# ...
```

Showing Details for All Namespaces

Use show namespace all to see details about all configured namespaces:

```
show namespace all
```

Showing Filer Shares Behind the Namespace

You can use show namespace mapping to list the filer shares that are behind all configured namespaces:

```
show namespace mapping
```

For example:

```
bstnA# show namespace mapping

Namespace               Physical Server
--------------------    ---------------------
insur:/claims
                        nas1:/vol/vol2/meta2*
                        \\nas1\insurance
                        \\nasE1\patient_records



Namespace               Physical Server
--------------------    ---------------------
labResearch:/labMice
                        \\fs3\mice_lab_results
                        \\fs4\white_mice
                        nas1:/vol/vol2/meta8*

labResearch:/labRats
                        \\fs6\labRats
                        \\fs7\lrTests
                        nas1:/vol/vol2/meta9*



Namespace               Physical Server
--------------------    ---------------------
medarcv:/lab_equipment
                        \\fs2\backlot_records
                        \\fs5\xraysScanners
                        nas1:/vol/vol2/meta6*
                        \\nas10\equipment
                        \\nas10\for_lease
                        \\nas11\equipBkup
                        \\nas11\leasedBkup

medarcv:/rcrds
                        \\fs1\histories
                        \\fs2\bulkstorage
                        \\fs4\prescriptions
                        nas1:/vol/vol2/meta3*

medarcv:/test_results
  2005charts            medarcv:/rcrds/2005
  chemLab               \\fs1\chem_results/.
  hematologyLab         \\fs3\hematology_results/.



Namespace               Physical Server
--------------------    ---------------------
medco:/vol
  vol1/corp             nas1:/vol/vol2/shr
  vol1/notes            nas1:/vol/vol2/notes
  vol2                 nas3:/exports/data
  vol3/mtgMinutes      nas2:/vol/datavol1/direct/mtgs
  vol3/sales           nas2:/vol/datavol1/direct/export


Namespace              Physical Server
--------------------   ---------------------
wwmed:/acct
                       das1:/exports/budget
                       das3:/exports/acct2
                       das7:/lhome/it5
                       das8:/work1/accting
                       nas1:/vol/vol2/meta1*


Where * denotes metadata only physical server.
bstnA# ...
```

Showing Shares Behind One Namespace

Add a namespace name to show only the shares behind that particular namespace:

show namespace mapping *name*

where ***name*** (1-30 characters) is the name of the namespace.

For example, this shows the filer shares behind the “wwmed” namespace:

```
bstnA# show namespace mapping wwmed

Namespace              Physical Server
--------------------   ---------------------
wwmed:/acct
                       das1:/exports/budget
                       das3:/exports/acct2
                       das7:/lhome/it5
                       das8:/work1/accting
                       nas1:/vol/vol2/meta1*


Where * denotes metadata only physical server.
bstnA# ...
```

Setting the Namespace Protocol(s)

The next step in creating a namespace is declaring the protocol(s) clients must use to access its data. All of the external filers in the namespace must support this entire protocol set; if the namespace supports both NFSv2 and NFSv3, all of its filer shares must also support both of those versions of NFS. Use the gbl-ns protocol command to identify one protocol for the namespace:

protocol {nfs2 | nfs3 | nfs3tcp | cifs}

where you must choose one of the flags, **nfs2**, **nfs3** (over UDP), **nfs3tcp** (NFSv3 over TCP) or **cifs**; these specify the protocol(s) supported by all of the namespace's filers. Repeat the command to specify multiple protocols for the namespace.

For example, this command set allows two forms of NFS access to the "wwmed" namespace:

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# protocol nfs2
bstnA(gbl-ns[wwmed])# protocol nfs3
bstnA(gbl-ns[wwmed])# ...
```

Removing a Protocol

Use the no form of the protocol command to remove a protocol from the namespace.

```
no protocol {nfs2 | nfs3 | nfs3tcp | cifs}
```

where you must choose one of the flags, **nfs2**, **nfs3** (over UDP), **nfs3tcp** (NFSv3 over TCP) or **cifs**.

For example, this command set removes NFSv2 from the "wwmed" namespace:

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# no protocol nfs2
bstnA(gbl-ns[wwmed])# ...
```

Creating a Multi-Protocol (NFS and CIFS) Namespace

If all the back-end shares support both NFS and CIFS, you can configure a *multi-protocol namespace*. Clients can access the same files from either a CIFS or an NFS client.

The namespace can be backed by a heterogeneous mix of multi-protocol filers, possibly from multiple vendors. The switch passes client requests to these filers, and passes filer responses back to the clients. File attributes, such as file ownership and permission settings, are managed by each filer. Each filer also manages its file and directory naming; if a name is legal in NFS but illegal in CIFS, each filer creates a filer-generated name (FGN) for its CIFS clients. Different vendors use different conventions for attribute conversions and FGNs, so that a CIFS-side name and/or ACLs at one filer may be different at another filer.

These issues are important to managed-volume configuration, and are discussed at length in the managed-volume chapter (as well as the *ARX CLI Maintenance Guide*).

Changing Protocols After Import

We strongly recommend that you choose your protocol set carefully, before configuring any volumes in the namespace. After a managed volume and at least one of its shares is enabled (as described later in the managed-volume chapter), the managed volume imports files and directories from its enabled shares. Protocol changes after this first import require greater care, since they may affect client access to the volume. The following changes affect client service:

- converting a single-protocol namespace to a multi-protocol namespace,
- removing CIFS, or
- removing all support for one version of the NFS protocol (that is, removing NFS2 or removing both of NFS3 and NFS3/TCP).

To make these conversions, you take the managed volume offline (with NSCK destage, described in the *ARX CLI Maintenance Guide*), change the protocol with this command, and restart the imports by re-enabling the managed volume.

For other protocol changes, such as adding NFSv2 to a running NFSv3 namespace, you must disable the volume (using no enable) and its front-end NFS service (described in a later chapter), change the protocol, then re-enable both. This causes a shorter service outage.

Setting Up Character Encoding

The next step in setting up a namespace is determining its *character encoding*, the encoding used for each character in its file names, directory names, and/or service names. This is the mapping of binary numbers to character symbols. ASCII is an example of a character encoding used commonly in the United States. NFS character encoding is discussed here, and character encoding for CIFS is discussed in *Verifying That a Service Can Support SMB2*, on page 11-49, as WINS name encoding.

NFS-character encoding applies to NFS file and directory names. A site has a character encoding for NFS file names that is shared among all clients and servers. This should be well-established before the installation of the ARX.

◆ **Note**

If there are multiple NFS namespaces associated with a single VIP, the NFS character encoding must be the same for all of those namespaces.

From gbl-ns mode, use the character-encoding nfs command to set the character encoding for NFS file names:

```
character-encoding nfs {utf-8 | iso-8859-1 | shift-jis | cp932 |
euc-jp | ksc5601}
```

where:

utf–8 specifies UTF–8 (Unicode, multi-byte) character encoding.

iso–8859–1 is ISO 8859–1 (Latin1, single-byte) character encoding.

shift-jis is Japanese character encoding, based on the JIS X 0208 Appendix 1 standard.

cp932 is Code Page 932, or Windows-31J (Japanese) character encoding. This is the Microsoft version of Shift_JIS.

euc-jp is variable-width Japanese character encoding, based on the JIS X 0208, JIS X 0212, and JIS X 0201 standards.

ksc5601 is Korean character encoding.

The default is ISO 8859–1 (Latin1).

For example, this command sequence sets character encoding to “UTF-8” (Unicode) for NFS file names in the ‘insur’ multi-protocol namespace:

```
bstnA(gbl)# namespace insur
bstnA(gbl-ns[insur])# protocol cifs
bstnA(gbl-ns[insur])# protocol nfs3
bstnA(gbl-ns[insur])# protocol nfs3tcp
bstnA(gbl-ns[insur])# character-encoding nfs utf-8
bstnA(gbl-ns[insur])# ...
```

Presumably, all of the multi-protocol filers behind the ‘insur’ namespace also support UTF-8 for their NFS file names.

Returning to Default Character Encoding For NFS

Use no character-encoding nfs to revert to ISO 8859–1 for NFS file names:

```
no character-encoding nfs
```

For example:

```
bstnA(gbl)# namespace insur
bstnA(gbl-ns[insur])# no character-encoding nfs
bstnA(gbl-ns[insur])# ...
```

You cannot change the character encoding after any of the namespace's managed volumes are enabled, as described in a later chapter.

Configuring CIFS Authentication

This section applies only to a namespace that supports CIFS. Skip to the next section if this is an NFS-only namespace.

The first step in configuring CIFS authentication is choosing one or more protocols for the namespace's clients. From gbl-ns mode, use the cifs authentication command to choose Kerberos, NTLM, or NTLMv2:

```
cifs authentication {kerberos | ntlm | ntlmv2}
```

> where **kerberos | ntlm | ntlmv2** is a required choice. If you choose NTLMv2, you must also verify that the *ARX Secure Agent* on your DC supports it. A section below describes the ARX Secure Agent and provides a link to upgrade instructions, if needed.

Re-run the command with a different option if you want the namespace to support additional authentication protocols. For example, this command set applies all three protocols to the "medarcv" namespace:

```
bstnA(gbl)# namespace medarcv
bstnA(gbl-ns[medarcv])# cifs authentication ntlm
bstnA(gbl-ns[medarcv])# cifs authentication ntlmv2

% INFO: To use NTLMv2 with the Acopia Secure Agent, all agent
instances must support NTLMv2 (requires agent version 5.1.0 or later).

bstnA(gbl-ns[medarcv])# cifs authentication kerberos
bstnA(gbl-ns[medarcv])# ...
```

If multiple CIFS namespaces are associated with a single VIP interface, the same CIFS authentication protocols must be used for all.

Removing an Authentication Protocol

Use the no form of the above command to stop supporting the given authentication protocol:

```
no cifs authentication {kerberos | ntlm | ntlmv2}
```

For example, these commands apply two authentication protocols to the "insur_bkup" namespace, and then later remove NTLM support. At the end of the sequence, the "insur_bkup" namespace only supports Kerberos authentication:

```
prtlndA(gbl)# namespace insur_bkup
prtlndA(gbl-ns[insur_bkup])# cifs authentication kerberos
prtlndA(gbl-ns[insur_bkup])# cifs authentication ntlm
...
prtlndA(gbl)# namespace insur_bkup
prtlndA(gbl-ns[insur_bkup])# no cifs authentication ntlm
prtlndA(gbl-ns[insur_bkup])# ...
```

Accessing NTLM Authentication Statistics

You can display statistics related to NTLM authentication on a particular domain controller using the CLI command show statistics domain-controller. The syntax for this command is:

The syntax is:

```
show statistics domain-controller ip-address
```

where *ip-address* specifies the domain controller for which you want to view the authentication statistics.

An example of the output from this command follows:

```
bstnA# show statistics domain-controller 192.168.25.102


************** Domain Controller NTLM authentications statistics **************

Domain Controller IP : 192.168.25.102
Authentication statistics since last reset at 07/15/2010 02:20:40 -0400
  Successful NTLM authentications   :       2
  Successful NTLMv2 authentications :       0
  Failed NTLM authentications       :       0
    No Such User                    :       0
    Bad Password                    :       0
    Account Disabled                :       0
    No logon server                 :       0
    Other                           :       0
  Failed NTLMv2 authentications     :       0
    No Such User                    :       0
    Bad Password                    :       0
    Account Disabled                :       0
    No logon server                 :       0
    Other                           :       0
  Schannel Inits                    :       1
  Avg Request SRT                   :       0 ms
  Min Request SRT                   :       0 ms
  Max Request SRT                   :       1 ms
  Avg DC SRT                        :       0 ms
  Min DC SRT                        :       0 ms
  Max DC SRT                        :       0 ms
```

You can reset the NTLM authentication statistics using the CLI command clear statistics domain-controller. The syntax for this command is:

```
clear statistics domain-controller {ip-address | all}
```

where *ip-address* specifies a particular domain controller for which you want to remove the authentication statistics, or all indicates that you want to remove the statistics for all domain controllers known to the current ARX.

For example, the following command clears the NTLM authentication statistics for the domain controller with the IP address 192.168.25.102:

```
bstnA# clear statistics domain-controller 192.168.25.102
```

Choosing a CIFS Proxy User

The next step in configuring CIFS authentication is declaring a *proxy user* for the namespace. A proxy-user is a valid Windows username, password, and domain that the namespace software can use as its identity. The namespace uses these credentials for operations that do not directly involve a Windows client, such as initial inventory of the CIFS shares and migration of files from one share to another. (Migration is a method for enforcing several namespace policies, which are described in later chapters.)

You must configure a proxy user for the namespace's domain ahead of time: the steps were described earlier in *Adding a Proxy User*, on page 3-4. The proxy user *must* belong to the Backup Operators group for all CIFS filers in the namespace. You can use the proxy user in probe exports to test this; this was described in *Probing for CIFS Security*, on page 5-15.

◆ **Note**

For filers where the namespace will access shares' paths, the proxy user must belong to the more-privileged Administrators group. This is required for supporting CIFS subshares, described later in this manual. To confirm that a proxy user belongs to a filer's Administrators group, use the proxy user in the show exports ... paths *command; this command was described in Showing the Physical Paths for CIFS Shares, on page 5-8.*

Each namespace can have one proxy user. The proxy-user credentials must be valid in the same Windows domain as the namespace. Multiple namespaces in the same domain can use the same proxy user. From gbl-ns mode, use the proxy-user command to apply a proxy user to the namespace:

proxy-user *name*

> where ***name*** (1-32 characters) identifies the proxy user for this namespace. Use the show proxy-user command for a list of configured proxy users.

For example, this command set applies a proxy user, "acoProxy2," to the "medarcv" namespace:

```
bstnA(gbl)# namespace medarcv
bstnA(gbl-ns[medarcv])# show proxy-user

Name             Windows Domain                Pre-Win2k        User Name
                 Description
--------------------------------------------------------------------------------
acoProxy1        WWMEDNET.COM                  WWMEDNET         jqprivate
                 jq's admin account

acoProxy3        FDTESTNET.COM                 BOSTONCIFS       jqtester

cifs_admin       MEDARCH.ORG                   MEDARCH          Administrator

nas_admin                                                       root

ny_admin         NY.COM                        NY               jqpublic

acoProxy2        MEDARCH.ORG                   MEDARCH          jqpublic
                 user with backup and admin creds on our servers
```

```
bstnA(gbl-ns[medarcv])# proxy-user acoProxy2
bstnA(gbl-ns[medarcv])# ...
```

Using Kerberos for Proxy-User Authentication

If the namespace's proxy-user configuration has an FQDN (such as "users.groups.org" instead of "users") for its Windows-domain, the namespace software uses Kerberos to access its back-end filers. Recall *Specifying the Windows Domain*, on page 3-4.

Opening Windows-Management Access (optional, MMC/Snapshots)

You can configure a group of Windows clients to have management authorization in this namespace, typically through the Microsoft Management Console (MMC). These clients can use MMC to make CIFS shares, open files, and/or any open CIFS-client sessions. In a managed volume, they may also be able to access volume snapshots. You configure this group of clients and their permissions ahead of time, as described in *Authorizing Windows-Management (MMC/Snapshot) Access*, on page 3-25. The group is called a Windows-management-authorization (*WMA*) group.

Each namespace can allow multiple WMA groups. From gbl-ns mode, use the windows-mgmt-auth command to apply one such group to the namespace:

```
windows-mgmt-auth name
```

where ***name*** (1-64 characters) identifies one WMA group for this namespace. Use the show windows-mgmt-auth command for a list of all configured WMA groups.

Enter this command once for each authorized group.

For example, this command set applies three WMA groups to the "medarcv" namespace:

```
bstnA(gbl)# namespace medarcv
bstnA(gbl-ns[medarcv])# show windows-mgmt-auth
...
bstnA(gbl-ns[medarcv])# windows-mgmt-auth testGroup
bstnA(gbl-ns[medarcv])# windows-mgmt-auth fullAccess
bstnA(gbl-ns[medarcv])# windows-mgmt-auth readOnly
bstnA(gbl-ns[medarcv])# ...
```

Removing a WMA Group

Use no windows-mgmt-auth to remove one WMA group from the current namespace:

```
no windows-mgmt-auth name
```

where ***name*** (1-64 characters) identifies one WMA group to remove from the namespace.

For example, this command sequence deletes the “testGroup” from the “medarcv” namespace:

```
bstnA(gbl)# namespace medarcv
bstnA(gbl-ns[medarcv])# no windows-mgmt-auth testGroup
bstnA(gbl-ns[medarcv])# ...
```

Selecting a SAM-Reference Filer

CIFS clients, given sufficient permissions, can change the users and/or groups who have access to a given file. For example, the owner of the “penicillin.xls” file can possibly add “nurses” or “doctors” to the list of groups with write permission. The list of groups in the network is traditionally provided by the Security Account Management (SAM) database on the file’s server. The ARX does not provide an internal SAM database; the namespace proxies all SAM queries to one of its CIFS filers, chosen at random.

◆ **Note**

According to the CIFS protocol, a CIFS-client application sends all of its SAM queries through a special pseudo share, IPC$. Each CIFS namespace therefore has a separate pseudo volume that it shares as IPC$. Since the queries come to this volume instead of the file’s volume, the namespace software does not know the file’s volume. Therefore, the namespace cannot intelligently choose an appropriate back-end filer as its SAM reference.

The filer used for SAM queries must contain a superset of all groups in all volumes, or some of the groups will be missing from the list. If any volume has filers that support Local Groups (as opposed to groups defined at the DC), you must configure one filer with all groups. If none of the filers use local groups, you can skip to the next section; the namespace can choose any of the filers as its SAM reference.

Use the gbl-ns sam-reference command to identify the pre-configured filer:

```
sam-reference filer
```

where ***filer*** (1-64 characters) is the external filer’s name, as displayed in show external-filer (see *Listing External Filers*, on page 6-18).

In the case of a disaster recovery configuration, an optional cluster argument can be appended to the command to specify the cluster with which the external filer is associated. In this case, the command syntax is:

```
sam-reference filer cluster clustername
```

where *clustername* specifies the cluster with which *filer* is associated.

If multiple CIFS namespaces are associated with a single VIP interface, the same SAM reference filer should be used for all of them.

For example, the following command sequence uses the “fs2” filer as a SAM reference for all volumes in the medarcv namespace:

```
bstnA(gbl)# show external-filer
  Name                        IP Address     Description
  ------------------------    -------------  ----------------------------
  das1                        192.168.25.19  financial data (LINUX filer, rack 14)
  fs1                         192.168.25.20  misc patient records (Table 3)
  fs2                         192.168.25.27  bulk storage server (Table 3)
  fs3                         192.168.25.28  Hematology lab server (Table 8)
  fs4                         192.168.25.29  prescription records (Table 3)
  fs5                         192.168.25.71  docs, invoices, for scanners (Table 7)
  fs6                         192.168.25.30  records of lab animals - lab rats (rack C)
  fs7                         192.168.25.41  lab animal test results - lab rats (rack C)
  das2                        192.168.25.22  Solaris filer 2 (rack 16)
  das3                        192.168.25.23  Solaris filer 3 (rack 16)
  nas1                        192.168.25.21  NAS filer 1 (rack 31)
                              192.168.25.61   (1st secondary)
                              192.168.25.62   (2nd secondary)
  das7                        192.168.25.24  Redhat-LINUX filer 1
  das8                        192.168.25.25  Redhat-LINUX filer 2
  nas2                        192.168.25.44  NAS filer 2 (rack 31)
  nas3                        192.168.25.47  NAS filer 3 (rack 32)
  nas10                       192.168.25.49  NAS filer 10 (rack 38)
  nas11                       192.168.25.48  filer 11 (rack 38)
  nasE1                       192.168.25.51  NAS filer E1
  smb1                        192.168.25.48  Samba filer
bstnA(gbl)# namespace medarcv
bstnA(gbl-ns[medarcv])# sam-reference fs2
bstnA(gbl-ns[medarcv])# ...
```

Using SMB Signing with Back-End Filers (optional)

SMB signing is the process of placing a digital signature in each Server Message Block (SMB) exchanged between a client and a server. You can use SMB signing between namespace software (in the role of a CIFS client) and its back-end filers (CIFS servers). This prevents man-in-the-middle attacks between the ARX and its filers, but at the cost of a performance penalty. This has no effect on interactions between CIFS clients and the namespace; you can enable SMB signing with clients later, when you create front-end services.

From gbl-ns mode, use the cifs filer-signatures command to enable SMB signing between this namespace and its filers:

```
cifs filer-signatures [required]
```

> where **required** (optional) makes the namespace refuse to connect to any filer that does not support SMB signing. If you omit this option, the namespace always attempts to connect with SMB signing enabled, but connects to a back-end filer even if the filer does not support it.

For example, this command sequence enables SMB signing for the medarcv namespace, but does not require it:

```
bstnA(gbl)# namespace medarcv
bstnA(gbl-ns[medarcv])# cifs filer-signatures
bstnA(gbl-ns[medarcv])# ...
```

Stopping SMB Signing

Use no cifs filer-signatures command to stop using SMB signing between the current namespace and its back-end filers:

```
no cifs filer-signatures
```

This breaks all CIFS communication with any filers that require SMB signing.

For example, this command sequence stops SMB signing between the “labResearch” namespace and its filers:

```
bstnA(gbl)# namespace labResearch
bstnA(gbl-ns[labResearch])# no cifs filer-signatures
bstnA(gbl-ns[labResearch])# ...
```

Permitting Anonymous Access (for Scanners)

Some scanners and copiers offer a “Save As” feature for saving scanned images to a remote CIFS share. This requires anonymous access to the IPC$ share on the target server. The scanner or copier uses this access to query the server about its CIFS shares and other capabilities. Each CIFS-supporting namespace has a virtual IPC$ “share” that can answer those queries. By default, anonymous clients are not allowed to access this “share” (or any CIFS volume) for security reasons.

Some CIFS-client operating systems (such as Mac OS X v10.4) also require this access for certain queries.

To support anonymous access to the IPC$ share, use the cifs anonymous-access command from gbl-ns mode:

```
cifs anonymous-access
```

This supports the “Save As” feature for certain scanners and copiers. See the ARX Release Notes for a list of supported models. With this option enabled, all volumes in the namespace support anonymous queries from these copiers and scanners.

If multiple CIFS namespaces are associated with a single VIP interface, cifs anonymous-access must be configured the same or all of them.

◆ **Note**

This permits the scanner to make its preliminary queries, but the scanner must provide a valid Windows username and password to actually save a file in the namespace.

For example, the following command sequence allows anonymous access to the IPC$ share in the “swic” namespace:

```
prtlndA(gbl)# namespace swic

This will create a new namespace.

Create namespace 'swic'? [yes/no] yes
prtlndA(gbl-ns[swic])# cifs anonymous-access
prtlndA(gbl-ns[swic])# ...
```

Denying Anonymous Access

For sites that do not support the “Save As” feature from scanners, you can use the no cifs anonymous-access command to deny anonymous access to the IPC$ volume:

```
cifs anonymous-access
```

This prevents support for scanners and photocopiers.

For example, the following command sequence stops anonymous access to the IPC$ share in the “medarcv” namespace:

```
bstnA(gbl)# namespace medarcv
bstnA(gbl-ns[medarcv])# no cifs anonymous-access
bstnA(gbl-ns[medarcv])# ...
```

Allowing SMB2 on the ARX (CIFS)

This section applies only to a namespace that supports CIFS. Skip to the next section if this is an NFS-only namespace.

SMB2 is a newer, widely-supported, version of the original CIFS (SMB) protocol. SMB2 support is disabled by default for the ARX. CIFS clients configured for SMB2 can only negotiate SMB support, not SMB2, with any ARX CIFS service. (CIFS services are described in a later chapter.) The ARX's internal processes (such as the policy engine) only use SMB2 for probes and queries that require it.

To enable SMB2 for the ARX and allow full use of SMB2, use the smb2-allowed command from gbl mode:

```
smb2-allowed
```

For example, the following command sequence offers SMB2 service on the "bstnA" ARX:

```
bstnA(gbl)# smb2-allowed
bstnA(gbl)# ...
```

Disabling SMB2

You can use the no smb2-allowed command to stop SMB2 access to all CIFS services and namespaces:

```
no smb2-allowed
```

For example, the following command sequence disables SMB2 on the "prtlndA" ARX:

```
prtlndA(gbl)# no smb2-allowed
prtlndA(gbl)# ...
```

You can also use this command in gbl-cifs mode (described in a later chapter) to disable SMB2 in one particular CIFS service.

Adding a Volume

The next, most important step in creating a namespace is creating one or more *volumes* for it. Each volume in a namespace is like a single file system. The volume aggregates one or more exports/shares from actual filers.

There are three forms of namespace volume: managed, direct, and shadow. The default type is a *managed volume*, so-called because it keeps metadata on all of the shares behind it and uses the metadata to manage the shares. A *direct volume* (called a *presentation volume* in the GUI) does not have any metadata or support policy. A *shadow volume* is a read-only copy of one or more managed volumes.

The next two chapters provide detailed instructions for configuring direct and managed volumes. Shadow volumes are described in a chapter after the storage-policy chapters. One namespace can support all three types of volumes.

From gbl-ns mode, use the volume command to create a volume:

```
volume topdir
```

where ***topdir*** (1-256 characters) is the directory with the contents of the filer shares. Volumes cannot contain one another: if you make a “/var” volume, you cannot have a “/var/log” volume in the same namespace; if you make a “/” volume, you cannot have any other volumes.

For a new volume, the CLI prompts for confirmation before adding it to the namespace. Enter **yes** to proceed. This puts you into gbl-ns-vol mode, where you must declare at least one share.

For example, this command set creates a single volume (“/acct”) for the “wwmed” namespace:

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# volume /acct
This will create a new volume.

Create volume '/acct'? [yes/no] yes
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Volume configuration is the bulk of namespace configuration. As mentioned above, the next chapters explain how to configure and maintain each volume type.

Enabling the Namespace (optional)

The final step in configuring a namespace is to enable it. This is optional, as each enabled volume in the namespace can process client requests independently. This command serially enables every volume in the namespace.

From gbl-ns mode, use the enable command to enable the current namespace:

```
enable
```

If the namespace contains any managed volumes, this causes all of them to start importing their enabled shares.

For example, the following command sequence enables the namespace named "wwmed:"

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# enable
bstnA(gbl-ns[wwmed])# ...
```

All imports happen asynchronously, so that you can issue more CLI commands while the imports happen in the background. You can use show namespace [status] to check the progress of the imports; this is discussed later in the managed-volume chapter.

◆ **Important**

For volumes backed by NetApp or EMC, you may need to access those filers directly and pre-create some qtrees or EMC filesystems. This rare configuration issue only occurs if:

- this is a managed volume,
*- you want to support both free-space quotas (*freespace cifs-quota*), and*
- you also want to support filer-subshares in this volume.

In this case, pre-create the NetApp qtrees and/or EMC filesystems for every subshare before you enable the share.

Enabling All Shares in the Namespace

From gbl-ns mode, you can enable all of the namespace's shares with a single command. Use the enable shares command to accomplish this:

```
enable shares
```

For example, the following command sequence enables all shares in the "wwmed" namespace:

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# enable shares
bstnA(gbl-ns[wwmed])# ...
```

Taking Ownership of All Managed Shares (optional)

Before a managed volume imports each share, it checks the root directory in the share for a special file that marks it as "owned" by an ARX. If this marker file exists, the managed volume does not proceed with the import; no two volumes can manage the same share. You may need to override this safety mechanism for a special case.

Consider an installation that uses legacy, filer-based applications to prepare for disaster recovery: it copies all of its directories and files from a primary site to another site. If an ARX manages the directories at the primary site, it places its ownership marker in the root of each share. The filer-based application copies the marker files to the remote site, along with all data files. An ARX at the backup site cannot import these shares because of the markers.

This does not apply to any direct volumes in the namespace.

You can use the optional take-ownership flag for this special case. If any managed volume finds an ownership marker in the root of a share, it overwrites the marker file. Otherwise, it imports the share as usual:

```
enable shares take-ownership
```

◆ **Important**

Do not use this option if it is possible that another ARX is managing one of the namespace's shares. This would unexpectedly remove the share(s) from service at the other ARX.

The CLI prompts for confirmation before taking ownership of any shares. Enter **yes** to proceed.

For example, the following command sequence enables all shares in the "insur_bkup" namespace and, if necessary, takes ownership of all of them:

```
prtlndA(gbl)# namespace insur_bkup
prtlndA(gbl-ns[insur_bkup])# enable shares take-ownership
This command allows the switch to virtualize shares that are used by other Acopia switches.
Allow switch to take ownership of all the shares in this namespace? [yes/no] yes
prtlndA(gbl-ns[insur_bkup])# ...
```

Disabling All Shares

Use no enable shares command to disable each of the namespace's individual shares:

```
no enable shares
```

For example, this command sequence disables all of the shares in the "ns1" namespace:

```
bstnA(gbl)# namespace ns1
bstnA(gbl-ns[ns1])# no enable shares
bstnA(gbl-ns[ns1])# ...
```

Disabling the Namespace

You can disable a namespace to stop clients from accessing it. This disables every volume in the namespace. Use no enable in gbl-ns mode to disable the namespace:

```
no enable
```

For example, the following command sequence disables the "wwmed" namespace:

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# no enable
bstnA(gbl-ns[wwmed])# ...
```

Showing Namespace Configuration

To review the configuration settings for a namespace, use the show global-config namespace command:

show global-config namespace [*name*]

> where ***name*** (optional, 1-30 characters) identifies the namespace. If you omit this, the output includes all namespaces

The output shows all of the configuration options required to recreate the namespace. The options are in order, so that they can be used as a CLI script.

For example, the following command shows the configuration for the "wwmed" namespace:

```
bstnA# show global-config namespace wwmed
;====================== namespace managed volumes =======================
namespace wwmed
  protocol nfs3
  protocol nfs3tcp
  description "namespace for World-Wide Medical network"
  volume /acct
    metadata critical
    modify
    policy migrate-method staged
    policy pause backupWindow
    reimport-modify
    reserve files 4000000
    metadata share nas1 nfs3 /vol/vol2/meta1
    share bills
      import sync-attributes
      policy freespace percent 2 resume-migrate 3
      critical
      filer das8 nfs /work1/accting
      enable
      exit

    share bills2
      import sync-attributes
      policy freespace percent 2 resume-migrate 3
      filer das3 nfs /exports/acct2
      enable
      exit

    share budget
      policy freespace percent 2 resume-migrate 3
      filer das1 nfs /exports/budget
      enable
      exit

    share it5
      import sync-attributes
      policy freespace percent 2 resume-migrate 3
      filer das7 nfs /lhome/it5
      enable
      exit

    share-farm fm1
      share bills
      share budget
      share bills2
      balance capacity
      auto-migrate
      enable
      exit

    place-rule docs2das8
      report docsPlc verbose
      inline report hourly docsPlc verbose
      from fileset bulky
      target share bills
      limit-migrate 50G
      enable
      exit

    volume-group 1
    enable
    exit

  exit

bstnA# ...
```

Removing a Namespace

From priv-exec mode, you can use the remove namespace command to remove a namespace and all of its volumes, as well as all front-end exports for the removed volumes:

```
remove namespace name [timeout seconds] [sync]
```

where:

name (1-30 characters) is the namespace to remove,

seconds (optional, 300-10,000) sets a time limit on each of the removal's component operations, and

sync (optional) waits for the removal to finish before returning. With this option, the CLI lists the namespace components as it removes them.

The CLI prompts for confirmation before removing the namespace. Enter **yes** to continue.

This operation generates a report, "removeNs_*namespace_date*.rpt," which catalogs all of the actions that it took. The *namespace* in the file name identifies the removed namespace, and the *date* is the date and time when the command started. The CLI shows the report name after you invoke the command. Use show reports to see the file listing; use show, tail, or grep to read the file. To save the report off to an external site, use the copy command from priv-exec mode.

The command does not create the report if you use the sync option; it shows its actions at the command line instead.

For example, this command sequence exits to priv-exec mode and then synchronously removes the insur_bkup namespace:

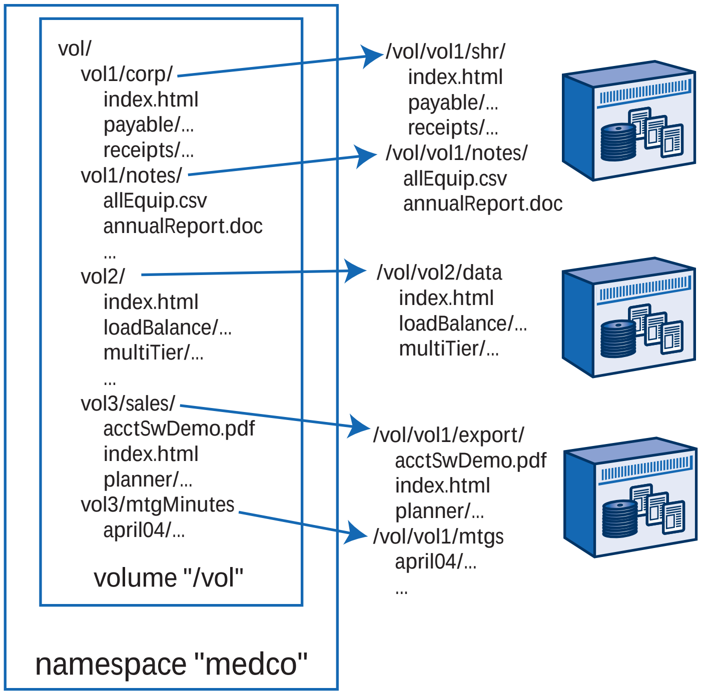
```
prtlndA(gbl)# end
prtlndA# remove namespace insur_bkup sync

Remove namespace 'insur_bkup'? [yes/no] yes
% INFO: Removing service configuration for namespace insur_bkup
% INFO: Removing CIFS browsing for namespace insur_bkup
% INFO: Removing volume policies for namespace insur_bkup
% INFO: destroy policy insur_bkup /insurShdw
% INFO: Removing shares for namespace insur_bkup
% INFO: no share backInsur
% INFO: Removing volume metadata shares for namespace insur_bkup
% INFO: no metadata share nas-p1 path /vol/vol1/mdata_B
% INFO: Removing volumes for namespace insur_bkup
% INFO: Removing NFS services for namespace insur_bkup
% INFO: Removing CIFS services for namespace insur_bkup
% INFO: no volume /insurShdw
% INFO: Removing namespace metadata shares for namespace insur_bkup
% INFO: Removing namespace insur_bkup
% INFO: no namespace insur_bkup
prtlndA# ...
```

8

Adding a Direct Volume

Overview

Each share in a *direct volume* attaches one or more of its own virtual directories to real directories at back-end shares. These *attach points* are analogous to mount points in NFS and network-drive connections in CIFS. The back-end directory trees are all reachable from the same volume root.The volume does not record the contents of the back-end shares, so it does not keep metadata or support storage policies. This is called a *presentation volume* in the GUI.



A direct volume is easier to set up than a managed volume, so this chapter is offered before the managed-volume chapter.

As explained earlier (in *Adding a Volume*, on page 7-22), you use the gbl-ns volume command to create a volume. This puts you into gbl-ns-vol mode, where you must declare this volume for use as a direct volume and create at least one direct share.

Note

Direct volumes do not support NFSv2. This protocol is set in the volume's namespace, as described earlier in Setting the Namespace Protocol(s), on page 7-8.

For example, this command set creates a single volume ("/vol") for the "medco" namespace:

```
bstnA(gbl)# namespace medco
bstnA(gbl-ns[medco])# volume /vol
This will create a new volume.

Create volume '/vol'? [yes/no] yes
```

```
bstnA(gbl-ns-vol[medco~/vol])# ...
```

Declaring the Volume “Direct”

A new volume is a managed volume by default; the next chapter describes the more-complex managed volumes. From gbl-ns-vol mode, use the direct command to convert a managed volume into a direct volume:

```
direct
```

For example, this command sequence converts the volume, “medco~/vol,” into a direct volume:

```
bstnA(gbl)# namespace medco
bstnA(gbl-ns[medco])# volume /vol
bstnA(gbl-ns-vol[medco~/vol])# direct
bstnA(gbl-ns-vol[medco~/vol])# ...
```

Reverting to a Managed Volume

If a direct volume has no attach points configured, you can use no direct to revert the volume back to a managed volume:

```
no direct
```

For example, this command sequence ensures that “wwmed~/acct” is a managed volume:

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# no direct
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

CLI Shortcut into gbl-ns-vol Mode

You can reach gbl-ns-vol mode from gbl mode by typing the name of the namespace and the volume in the same command. From gbl mode, you can use this syntax:

```
namespace namespace-name volume vol-path
```

where

namespace-name identifies the namespace, and

vol-path is the volume name.

For example, this command sequence reaches the “medco~/vol” volume with a single command:

```
bstnA(gbl)# namespace medco volume /vol
bstnA(gbl-ns-vol[medco~/vol])# ...
```

Setting the Volume’s Free Space (Optional)

The next step in creating a volume is to choose an algorithm for calculating its free space. This is the free space calculation that is passed on to the client: whenever a user mounts a volume (NFS) or maps a network drive to it (CIFS), this total is the free space that they see.

By default, the namespace volume’s free space is the sum of the free space on all of its back-end shares *except* shares from the same storage volume. If two or more shares report the same ID for their backing volume, only the first share is counted. (For NFS shares, the volume uses the file system ID (FSID); for CIFS shares, it uses the Volume Serial Number). If this default is acceptable, you can skip the rest of this section.

Using Manual Free Space Calculation

You may wish to control the free space calculation manually on a share-by-share basis. This means counting all free space on all shares, regardless of duplicate back-end-volume IDs, then ignoring certain shares manually or adjusting their free space reporting. You can use the freespace ignore and freespace adjust commands, described later, to ignore the free space from a share or change the reported value for the free space.

From gbl-ns-vol mode, use the freespace calculation manual command to override the default free space calculation:

```
freespace calculation manual
```

You can set this at any time, even after the volume is enabled.

For example, this command sequence makes the ‘access~/G’ volume count the free space in all back-end shares, even multiple shares that draw from the same back-end storage:

```
prtlndA(gbl)# namespace access volume /G
prtlndA(gbl-ns-vol[access~/G])# freespace calculation manual
prtlndA(gbl-ns-vol[access~/G])# ...
```

Using Automatic Free Space Calculation (Returning to Default)

By default, free space is calculated based on the IDs of the volumes behind the back-end shares. If any of these shares report the same volume ID, their free space is counted only once. To return to this default, use the no freespace calculation manual command:

```
no freespace calculation manual
```

For example:

```
bstnA(gbl)# namespace medco volume /vol
bstnA(gbl-ns-vol[medco~/vol])# no freespace calculation manual
bstnA(gbl-ns-vol[medco~/vol])# ...
```

Setting CIFS Options

The next step in configuring a volume is addressing its CIFS options, if necessary: This section does not apply to volumes in NFS-only namespaces; skip to the next section if this namespace does not support CIFS.

There are five CIFS-volume attributes that back-end filers may or may not support. They are named streams, compressed files, persistent ACLs, Unicode file names on disk, and sparse files. Each volume can support any and all of these capabilities. However, *all* filer shares used in a volume *must* support the capabilities advertised in the volume. For example: if your volume is going to support compressed files, then *all* of its back-end filer shares must also support compressed files.

By default, new volumes conform to the CIFS options at the first-enabled share. If you try to enable another volume share that does not support one or more of those options, the you get an error and the share remains disabled. You can then disable the unsupported options (the error message tells you which ones) and retry the enable. For an earlier view of all CIFS options on the filer, you can use the show exports command; refer back to *Showing CIFS Attributes*, on page 5-13.

From gbl-ns-vol mode, use any combination of the following gbl-ns-vol commands to manually set the options:

```
[no] named-streams
```

```
[no] compressed-files
```

```
[no] persistent-acls
```

```
[no] unicode-on-disk
```

```
[no] sparse-files
```

For example, the following command sequence disables named streams and sparse files in the “medarcv~/test_results” volume:

```
bstnA(gbl)# namespace medarcv volume /test_results
bstnA(gbl-ns-vol[medarcv~/test_results])# no named-streams
bstnA(gbl-ns-vol[medarcv~/test_results])# no sparse-files
bstnA(gbl-ns-vol[medarcv~/test_results])# ...
```

As another example, this command sequence reinstates named streams in the same volume:

```
bstnA(gbl)# namespace medarcv volume /test_results
bstnA(gbl-ns-vol[medarcv~/test_results])# named-streams
bstnA(gbl-ns-vol[medarcv~/test_results])# ...
```

Advertising Access-Based-Enumeration (ABE) Support

Some CIFS filers support *Access-Based Enumeration* (ABE), meaning that their clients only see the files for which they have read-access permissions. That is, if a folder contains files or sub folders that a client does not have permission to read, those files and sub folders do not appear in the client's view. This is a security feature designed to reduce client curiosity about inaccessible files and folders.

The back-end filers behind the direct volume support ABE independently. If a back-end filer supports ABE, it does not display inaccessible files to the direct volume's clients. Filers without ABE display all files and directories, regardless of access permissions.

The Windows *abecmd.exe* utility, which is designed to enable or disable ABE on remote servers, can query the volume for whether or not ABE is enabled. By default, a direct volume answers this query in the negative, declaring that ABE is disabled. To make the volume answer in the affirmative, go to gbl-ns-vol mode and use the cifs access-based-enum command:

```
cifs access-based-enum
```

For example, the following command sequence advertises the ABE feature in the "medarcv~/test_results" volume:

```
bstnA(gbl)# namespace medarcv volume /test_results
bstnA(gbl-ns-vol[medarcv~/test_results])# cifs access-based-enum
bstnA(gbl-ns-vol[medarcv~/test_results])# ...
```

CIFS clients see the results of this command only if they connect after you invoke it.

Disabling ABE

If none of the filers behind this volume have ABE enabled, the volume should tell the abecmd.exe utility that ABE is disabled. You can use the no cifs access-based-enum command (the default) to stop the current volume from claiming that ABE is enabled:

```
no cifs access-based-enum
```

For example, the following command sequence makes the "medarcv~/test_results" volume declare that ABE is disabled:

```
bstnA(gbl)# namespace medarcv volume /test_results
bstnA(gbl-ns-vol[medarcv~/test_results])# no cifs access-based-enum
bstnA(gbl-ns-vol[medarcv~/test_results])# ...
```

Adding a Share

The next step in creating a direct volume is identifying one or more shares for it. A *share* maps to a single CIFS share or NFS export from an external (NAS or DAS) filer. A *direct share* contains one or more attach-point directories, each attached to a real directory on its back-end share. A direct volume can contain dozens of shares, where each could possibly correspond to a different filer.

From gbl-ns-vol mode, use the share command to add a share to a volume:

```
share name
```

where ***name*** (1-64 characters) is a name you choose for the new share.

As with new namespaces and volumes, the CLI prompts for confirmation before creating a new share. Enter **yes** to proceed. This puts you into gbl-ns-vol-shr mode, where you must identify the filer and export/share, and then you must enable the namespace share.

For example, this command set adds a share called “corporate” to the “/vol” volume in the “medco” namespace:

```
bstnA(gbl)# namespace medco volume /vol
bstnA(gbl-ns-vol[medco~/vol])# share corporate
This will create a new share.

Create share 'corporate'? [yes/no] yes
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# ...
```

CLI Shortcut into gbl-ns-vol-shr Mode

You can reach gbl-ns-vol-shr mode from gbl mode by typing the name of the namespace, volume, and share in the same command. From gbl mode, you can use this syntax:

```
namespace namespace-name volume vol-path share share-name
```

where

namespace-name identifies the namespace,

vol-path is the volume name, and

share-name is the share.

For example, this command sequence reaches the “medco~/vol~corporate” share with a single command:

```
bstnA(gbl)# namespace medco volume /vol share corporate
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# ...
```

Listing Filer Shares

It is convenient to show the available back-end-filer shares as you add them into a direct volume. Filer shares are configured from Network-Attached Storage (NAS) filers or file servers with Direct-Attached Storage (DAS). Use the show exports external-filer ... shares command to show a filer’s shares using the external-filer name assigned at the ARX. This is very similar to the show exports host command described in *Chapter 5, Examining Filers*:

```
show exports external-filer filer shares
[user username windows-domain domain | proxy-user proxy]
```

where

filer (1-1024 characters) is the external filer's name, as displayed in show external-filer (see *Listing External Filers*, on page 6-18),

username (optional, 1-32 characters) is a Windows username (for CIFS shares),

domain (optional, 1-1024 characters) is the above user's Windows domain, and

proxy (1-32 characters) is an alternative to the username/domain credentials. This the name of the proxy-user configuration, as shown by the show proxy-user command (see *Listing All Proxy Users*, on page 3-7).

The output shows NFS shares if no user or proxy-user is specified, and CIFS shares if a user or proxy-user is specified:

- The NFS table shows each export and the NIS netgroup(s) (shown as IP addresses and wildcards) that can access the share. For an export to be eligible for inclusion in the volume, all proxy-IP addresses must be on this list.
- The CIFS table shows two disk-space measures and the serial number for the storage volume behind the share. If two shares have the same serial number, they are assumed to be shares for the same storage volume on the filer.

For example, the following command shows the NFS shares on the "das8" external filer:

```
bstnA(gbl)# show exports external-filer das8 shares
Export probe of filer "das8" at 192.168.25.25

Shares:
  NFS
                                                  Read   Write     Space
    Path (Owner)                         Status   Size   Size   Total  Avail   FSID   Access
    ---------------------------------    ------- ------ ------ ------ ------ -------- ------
    /work1                               Mounted  32 kB  32 kB  70 GB  27 GB fd00     *

bstnA(gbl)# ...
```

Showing Supported Protocols at the Filer

The filer's supported protocols must include all of the namespace protocols. To check them, use the capabilities keyword in the show exports command:

show exports external-filer *filer* capabilities

For example, this command shows that the "fs4" filer supports SMB (the originally-released version of CIFS) and SMB2.002:

```
bstnA(gbl)# show exports external-filer fs4 capabilities proxy-user acoProxy2
Export probe of filer "fs4" at 192.168.25.29

CIFS Credentials:
    User             jqpublic
    Windows Domain   MEDARCH.ORG
    Pre-Win2k Domain  MEDARCH

Capabilities:

  CIFS
    Server           TCP/445
    SMB:
      Security Mode   User level, Challenge/response, Signatures disabled
      Capabilities    Large Files, Large Read, Large Write, Info Passthru
      Max Buffer      16644
    SMB2.002:
      Security Mode   Signatures optional
      Max Read        65536
      Max Write       65536
      Max Transact    65536
    Auth Protocols    NTLMSSP, Kerberos
    IPC$ Share        Access OK
    Auth Method Used  Kerberos
    SPN Used          VM-SWP2008S-02@MEDARCH.ORG
    Discovered SPN    VM-SWP2008S-02@MEDARCH.ORG
bstnA(gbl)# ...
```

Identifying the Filer and Share

The next step in configuring a direct share is identifying its source share on an external filer. From gbl-ns-vol-shr mode, use the filer command to use a filer's exported directory as a share:

filer *name* {{nfs | cifs} *share-name*} [access-list *list-name*] [cluster *clustername*]

where

name (1-64 characters) is the name of the external filer (see *Listing External Filers*, on page 6-18),

nfs | cifs is a required choice,

share-name (1-900 for an NFS path, 1-1024 characters for a CIFS share) is the NFS export or CIFS share on the filer, and

list-name (optional, 1-64 characters) is the name of an NFS access list to associate with the share (see *Listing All NFS Access Lists*, on page 4-9).

clustername is the name of an ARX cluster with which the filer is associated in a disaster recovery configuration.

Never connect to more than one instance of a given back-end share. If the filer uses aliases for its shares, the ARX assumes that each alias is a different share.

Note

You can use an NFSv3 export from another ARX if (and only if) the export is backed by a managed volume, described in the next chapter. A direct volume cannot attach to another direct volume.

Important

If you use an NFS export from another ARX, its managed volume must have auto reserve files *disabled. The direct volume assumes that the number of reserved files in an NFS share is fixed; it cannot support an automatic increase in this limit. The next chapter describes the* no auto reserve files *command as it relates to this issue.*

For example, this command set identifies a filer share for the direct share, “corporate.” The share is /vol/vol1/direct on a filer named “nas1:”

```
bstnA(gbl)# namespace medco volume /vol share corporate
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# show exports external-filer nas1 shares
Export probe of filer "nas1" at 192.168.25.21

Shares:
  NFS
                                                  Read   Write     Space
    Path (Owner)                         Status    Size   Size   Total  Avail FSID     Access
    ---------------------------------    ------- ------ ------ ------ ------ -------- ------
    /vol/direct_scale                    Mounted  32 kB  32 kB 5.0 GB 4.9 GB b2008ca5
    /vol/vol0                            Mounted  32 kB  32 kB  22 GB  21 GB 90008ca5
    /vol/vol1                            Mounted  32 kB  32 kB  63 GB  58 GB 48008ca5
    /vol/vol1/direct                     Mounted  32 kB  32 kB  63 GB  58 GB 48008ca5
    /vol/vol10                           Mounted  32 kB  32 kB  56 GB  54 GB ae008ca5
    /vol/vol16                           Mounted  32 kB  32 kB  16 GB  16 GB c6008ca5
    /vol/vol2                            Mounted  32 kB  32 kB  17 GB  16 GB a8008ca5
    /vol/vol3                            Mounted  32 kB  32 kB  68 GB 3.9 MB a7008ca5
    /vol/vol4                            Mounted  32 kB  32 kB  68 GB  26 GB a5008ca5
    /vol/vol5                            Mounted  32 kB  32 kB  68 GB  39 GB a9008ca5
    /vol/vol6                            Mounted  32 kB  32 kB  67 GB  67 GB aa008ca5
    /vol/vol7                            Mounted  32 kB  32 kB  68 GB 146 MB ab008ca5
    /vol/vol8                            Mounted  32 kB  32 kB  68 GB 5.5 GB ac008ca5
    /vol/vol9                            Mounted  32 kB  32 kB  57 GB  27 GB ad008ca5
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# filer nas1 nfs /vol/vol1/direct
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# ...
```

Identifying a Multi-Protocol Share

In a multi-protocol (NFS and CIFS) namespace, you list both names for the share. You can do this in any order:

filer *name* nfs *nfs-name* cifs *cifs-name* [access-list *list-name*]

or

filer *name* cifs *cifs-name* nfs *nfs-name* [access-list *list-name*]

Disconnecting From the Filer

From gbl-ns-vol-shr mode, use the no filer command to disconnect the current direct share from its filer:

no filer

For example:

```
bstnA(gbl)# namespace medco volume /vol share test
bstnA(gbl-ns-vol-shr[medco~/vol~test])# no filer
```

Using a Managed Volume as a Filer

You can assign a managed volume to the share as though it were an external filer. (The next chapter describes how to configure a managed volume.)

Note

If the direct volume's namespace supports CIFS, you can only use a managed volume from the same namespace.

From gbl-ns-vol-shr mode, use the managed-volume command to use a managed volume as a direct share's "filer:"

managed-volume *namespace volume* [access-list *list-name*]

where:

namespace (1-30 characters) is the name of the namespace.

volume (1-1024 characters) is the volume's path (for example, '/rcrds'). You cannot use a direct volume; only a managed volume.

list-name (optional, 1-64 characters) is the NFS access list to associate with the share (see *Listing All NFS Access Lists*, on page 4-9).

For example, the following command sequence assigns the 'wwmed~/it' managed volume to the 'test' share.

```
bstnA(gbl)# namespace medco volume /vol share test
bstnA(gbl-ns-vol-shr[medco~/vol~test])# managed-volume wwmed /it
bstnA(gbl-ns-vol-shr[medco~/vol~test])# ...
```

Disconnecting from the Managed Volume

Use the no form of the command to stop using a managed volume as a direct share's filer.

```
no managed-volume
```

Attaching a Virtual Directory to the Back-End Share

The next step is to create a virtual *attach-point directory*, visible to clients from the root of the volume, and attach it to a physical directory on the back-end filer. For example, you can create an attach point named /vol1 (in the /vol volume) and attach it to the filer's /usr/local directory: the client-viewable /vol/vol1 is then the same as /usr/local on the filer. Clients can mount any level of the volume or attach-point directory (/vol or /vol/vol1). Mounting below that level (for example, /vol/vol1/work) defeats the purpose of federating the attach points, so it is not supported.

From gbl-ns-vol-shr mode, use the attach command to create an attach-point directory and attach it to one of the filer's physical directories:

```
attach attach-point-directory [to physical-directory] [access-list
list-name]
```

where:

attach-point-directory (1-256 bytes of UTF-8-encoded characters) is the path of the virtual directory that the client sees. This is relative to the root of the volume; for example, if you are in the /home volume and you enter a virtual directory of "aa," clients see this attach point as /home/aa.

to *physical-directory* (optional; 1-256 bytes of UTF-8-encoded characters) specifies the name of the actual directory on the filer share. Similar to the attach-point directory, this path is relative to the root of the filer share (established in the filer command's nfs or cifs clause, above). Use a "." to attach to the root directory in the share. If you omit this clause, the ARX uses the *attach-point-directory* path.

list-name (optional 1-64 characters) is the NFS access list to associate with this attach point (see *Listing All NFS Access Lists*, on page 4-9).

You can use the command multiple times in the same direct share, thereby attaching to multiple directories on the back-end share. The maximum number of total attach points is resource- and platform-dependent: a later section describes various limits on resource usage, including attach points.

Attach-point directories cannot be nested: if you create /var in one attach point, you cannot create /var/log in another attach point in the same volume.

For example, this command sequence sets up the filer for the "corporate" share (as shown above), then attaches three directories to the filer:

```
bstnA(gbl)# namespace medco volume /vol share corporate
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# filer nas1 nfs /vol/vol1/direct
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# attach vol1/corp to shr
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# attach vol1/notes to notes
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# attach conCalls
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# ...
```

This creates the following map of virtual to physical directories:

Virtual Directories (seen by clients)	Physical Directories (on filer)
/vol/vol1/corp	/vol/vol1/direct/shr
/vol/vol1/notes	/vol/vol1/direct/notes
/vol/conCalls	/vol/vol1/direct/conCalls

Removing an Attach Point

Use the no attach command to remove one attach point from the share:

```
no attach attach-point-directory
```

where ***attach-point-directory*** (1-256 bytes of UTF-8-encoded characters) is the path of the virtual directory to detach.

For example, this command sequence detaches the conCalls directory from the “corporate” share:

```
bstnA(gbl)# namespace medco volume /vol share corporate
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# no attach conCalls
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# ...
```

Designating the Share as Critical (Optional)

If the current switch has a redundant peer, you have the option to designate the current share as *critical*. Skip to the next section if this switch is not configured for redundancy.

If the direct volume software loses contact with one of its critical (and enabled) shares, the ARX initiates a failover. The failover is accepted by the redundant peer as long as the peer has full access to all critical shares, critical routes, *and* the quorum disk. If the peer is unable to access any of these critical resources, no failover occurs. (For instructions on configuring critical routes, refer back to *Identifying a Critical Route*, on page 7-18 of the *ARX® CLI Network-Management Guide*.)

From gbl-ns-vol-shr mode, use the critical command to designate the current share as a critical resource:

```
critical
```

For example, this command sequence designates the “corporate” share as a critical share:

```
bstnA(gbl)# namespace medco volume /vol share corporate
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# critical
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# ...
```

Removing Critical-Share Status

By default, shares are not critical. If the switch loses contact with a non-critical share, the volume operates in a degraded state and the switch does not initiate a failover. From gbl-ns-vol-shr mode, use the no critical command to make the current share non-critical:

```
no critical
```

For example, this command sequence removes the “generic” share from the list of critical shares:

```
bstnA(gbl)# namespace medco volume /vol share generic
bstnA(gbl-ns-vol-shr[medco~/vol~generic])# no critical
bstnA(gbl-ns-vol-shr[medco~/vol~generic])# ...
```

Ignoring the Share’s Free Space (Optional)

This option is relevant only in a volume where you are calculating free space manually (recall *Setting the Volume’s Free Space (Optional)*, on page 8-5). Such a volume’s free space is the sum of the space from *all* of its shares, including multiple shares from the same back-end storage volume. This can mean counting the same storage multiple times: two or more shares from the same storage volume each report the full amount of free space on the volume. For example, two NFS shares from the same disk partition, /lhome, would each report the total free space on the /lhome partition. A volume with both of these shares would double-count the free space in /lhome.

You can manually exclude shares from the free space calculation. From gbl-ns-vol-share mode, use the freespace ignore command to ignore the free space on the current share:

```
freespace ignore
```

You can set this before or after the share and volume are enabled.

For example, this command sequence ignores all free space in the “rec98” share:

```
prtlndA(gbl)# namespace access volume /G share rec98
prtlndA(gbl-ns-vol-shr[access~/G~rec98])# freespace ignore
prtlndA(gbl-ns-vol-shr[access~/G~rec98])# ...
```

Including the Share in the Free Space Calculation

By default, free space from all shares is counted toward the volume’s total. To include this share in the free space calculation, use no freespace ignore:

```
no freespace ignore
```

For example:

```
prtlndA(gbl)# namespace access volume /G share recsY2k
prtlndA(gbl-ns-vol-shr[access~/G~recsY2k])# no freespace ignore
prtlndA(gbl-ns-vol-shr[access~/G~recsY2k])# ...
```

Adjusting the Free Space Calculation

You can also manually adjust the free space that is advertised for the current share. From gbl-ns-vol-share mode, use the freespace adjust command:

```
freespace adjust [-]adjustment[K|M|G|T]
```

where:

- (optional) makes the adjustment negative,

adjustment is the size of the adjustment, and

K|M|G|T (optional) chooses the unit of measure: **K**ilobytes, **M**egabytes, **G**igabytes, or **T**erabytes. The default is bytes if you omit this. All values are base-2; e.g., a kilobyte is 1,024 bytes and a megabyte is 1,048,576 bytes.

As with freespace ignore, you can set this before or after the share and volume are enabled. If you combine this with freespace ignore, the value you set here is the total advertised free space for the share.

For example, this command sequence increases the free space calculation for the "recs2002" share:

```
prtlndA(gbl)# namespace access volume /G share recs2002
prtlndA(gbl-ns-vol-shr[access~/G~recs2002])# freespace adjust 1G
prtlndA(gbl-ns-vol-shr[access~/G~recs2002])# ...
```

Erasing the Free Space Adjustment

Use the no freespace adjust command to remove any free space adjustment from the current share:

```
no freespace adjust
```

For example, this command sequence removes any free space adjustment from the "corporate" share:

```
bstnA(gbl)# namespace medco volume /vol share corporate
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# no freespace adjust
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# ...
```

Enabling the Share

The final step in configuring a share is to enable it. An enabled share is an active part of the direct volume; clients can see all the share's attach-point directories after the share is enabled. To enable a share, use the enable command in gbl-ns-vol-shr mode:

```
enable
```

In the show namespace output, a direct share has a status of "Online: Direct." In show namespace status, the share's status is abbreviated to "Online."

For example, the following command sequence enables the medco ~/vol~corporate share.

```
bstnA(gbl)# namespace medco volume /vol share corporate
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# enable
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# ...
```

◆ **Note**

Executing this command with the optional take ownership *argument (that is,* enable take ownership*) causes you to take ownership of the share when it is in a volume used by another ARX.*

◆ **Important**

For volumes backed by NetApp or EMC, you may need to access those filers directly and pre-create some qtrees or EMC filesystems. This rare configuration issue only occurs if:

- this is a managed volume,
*- you want to support both free-space quotas (*freespace cifs-quota*), and*
- you also want to support filer-subshares in this volume.

In this case, pre-create the NetApp qtrees and/or EMC filesystems for every subshare before you enable the share.

Disabling the Share

You can disable a direct share to remove its attach points from the volume. This causes all of the share's attach-point directories to effectively disappear from client view. Use no enable in gbl-ns-vol-shr mode to disable the share:

no enable

For example, the following command sequence disables the medco~/vol~sales share.

```
bstnA(gbl)# namespace medco volume /vol share sales
bstnA(gbl-ns-vol-shr[medco~/vol~sales])# no enable
bstnA(gbl-ns-vol-shr[medco~/vol~sales])# ...
```

Changing the Maximum Files on the Back-End Share

The volume software records the maximum number of files on each direct share when you enable the share. To change the maximum, you must disable the share at the ARX, access the filer directly to change the maximum number of files, and then re-enable the share on the ARX.

Removing a Direct Share

Use the no share command to remove a share from a direct volume:

no share

For example, this command set removes the "test" share from the "/vol" volume in the "medco" namespace:

```
bstnA(gbl)# namespace medco volume /vol
bstnA(gbl-ns-vol[medco~/vol])# no share test
bstnA(gbl-ns-vol[medco~/vol])# ...
```

Assigning The Volume To A Volume Group (optional)

The next step in configuring a volume is to assign it to a volume group. The ARX's namespace functionality will assign the volume to a default volume group automatically if you do not specify one explicitly.

Volume groups are a means of isolating namespaces in the ARX's memory so that the failure of one or more namespaces in one volume group does not affect the performance of namespaces in other volume groups. Volume groups are a virtual resource that can fail over from one ARX to another in a redundant configuration.

The maximum number of volume groups supported per ARX model is a function of the ARX's memory, and can be between 2 and 24, depending on the ARX model. Within that range of possible volume groups is the number of volume groups that are allowed by the particular license agreement that the customer has in place for a specific ARX instance. This number of volume groups may be less than the maximum, depending on the number that the customer has paid to license, but will never exceed the maximum number supported for that model. Follow this link to a table that shows the maximum number of volume groups supported by each ARX model:

http://support.f5.com/content/dam/f5/kb/global/solutions/sol13129_images.html/Platform_matrix_ARX_system_limits_enforced_by_licensing.pdf

◆ **Note**

For the ARX-500, two volume groups are supported by default, and a CLI command can be used to increase the number of volume groups to the maximum number supported for that model, which is four. This is an artifact of the hardware design of this model.

Assigning a Volume To a Volume Group Explicitly

When you are configuring a volume and are in gbl-ns-vol mode, you can assign the volume to a specific volume group by using the volume-group CLI command. The syntax is:

```
volume-group volumegroupid
```

where *volumegroupid* is the integer that identifies the volume group to which you are assigning the current volume.

For example:

```
bstnA(gbl-ns-vol[bgh~/naumkeag_wing])# volume-group 2
```

assigns the current volume, “naumkeag_wing”, to volume group 2.

Migrating a Volume to a Different Volume Group

As your deployment grows, you may want to redistribute volumes among volume groups to mitigate possible down time. You can move one or more volumes from one volume group to another by using the nsck command with the migrate-volume option. This is done in privileged-exec mode.

The command syntax is:

```
nsck name migrate-volume volname volume-group id [check-limits]
```

where:

name is the namespace within which the volume will be moved;

volname specifies the volume to be moved; and

id specifies the volume-group to which the volume will be moved.

The check-limits argument is optional, and enables you to verify that the volume migration will work without actually initiating it. It checks the licensed limits at the destination volume group, and for any other possible rules that might prevent a successful migration.

The example that follows specifies a volume to migrate, but uses the check-limits option to verify it first:

```
bstnA# nsck medarcv migrate-volume /test_results volume-group 3
check-limits
```

Then, execute the command without the check-limits argument to initiate the migration of the volume to a different volume group:

```
bstnA# nsck medarcv migrate-volume /test_results volume-group 3
Volume /test_results is in use by CIFS global service
'ac1.medarch.org'.
Volume /test_results in namespace medarcv is in use or being browsed
by global services.
Migrating the volume to a different Volume-Group will disrupt all
clients accessing this volume through these services.
No other volume will be affected by this procedure.
Proceed with volume migration? [yes/no] yes
Scheduling report: migrate_volume_group.35.rpt on switch bstnA
```

To cancel the volume migration (and return the volume to service in its original volume group), use cancel migrate-volume.

The nsck migrate-volume command stops client access to that volume until the migration is complete. When the command is executed, the CLI lists all of the affected front-end services and prompts you for confirmation; type “yes” to continue with the migration.

You cannot migrate a volume while another nsck operation is in progress.

A volume cannot be migrated while it is importing files from its back-end shares. The import must be complete before you can move the volume to another volume group.

A volume group cannot be assigned if it is in use by another namespace, or if it is the target of a migration in progress.

Very large volumes may take a long time to finish migrating. Upon execution, the nsck migrate-volume command returns immediately, providing you with the name of a detailed report that will record the migration. The migration report is named "migrate_volume_group.*nsck-job-id*.rpt." Use show reports to list all reports, including this one. Use the tail reports *report-name* follow command to follow the progress of the migration. You can continue using the CLI while the migration occurs in the background. The show nsck command shows the current status of the migration, along with the status of all other nsck operations. After the nsck job is complete, front-end services will require a few additional seconds before they allow client access to the volume.

Reverting to The Default Volume Group Assignment

Prior to enabling a volume, you can revert it to its default volume group assignment by using the no volume-group CLI command in gbl-ns-vol mode:

```
bstnA(gbl-ns-vol[bgh~/naumkeag_wing])# no volume-group
```

This undoes any explicit volume group assignment that you may have configured, and assigns the volume to whatever default volume group it would have been assigned automatically.

Showing Volume Groups On The Current ARX

Use the show volume-group CLI command to display all of the volume groups on the current ARX. For example:

```
bstnA# show volume-group


Switch: bstnA
----------------------------------------------------------------------
System Credits
--------------
Share credits: 4 shares used (380 credits remain of total 384)
Direct share credits: 1 direct shares used (6143 credits remain of
total 6144)
Volume credits: 2 volumes used (94 credits remain of total 96)
File credits: 4.0 M files reserved (764 M credits remain of total 768
M)


Volume Group 1
------------
Physical Processor: 1.1
State: Normal; partially used
Share credits: 4 shares used (188 credits remain of total 192)
Direct share credits: 1 direct shares used (3071 credits remain of
total 3072)
```

```
Volume credits: 2 volumes used (46 credits remain of total 48)
File credits: 4.0 M files reserved (380 M credits remain of total 384
M)

Namespace         Volume Group  Volume              State
---------         ------------  ------              -----
bgh                1      /naumkeag_wing       Enabled

1 Namespace            1 Volume
```

Display a more detailed set of information for the volume groups by appending the detailed argument to the command:

```
bstnA# show volume-group detailed

Switch: bstnA
----------------------------------------------------------------------
System Credits
--------------
Share credits: 4 shares used (380 credits remain of total 384)
Direct share credits: 1 direct shares used (6143 credits remain of
total 6144)
Volume credits: 2 volumes used (94 credits remain of total 96)
File credits: 4.0 M files reserved (764 M credits remain of total 768
M)

Volume Group 1
------------
Physical Processor: 1.1 (29% CPU, 40% MEM)
State: Normal; partially used
Share credits: 4 shares used (188 credits remain of total 192)
Direct share credits: 1 direct shares used (3071 credits remain of
total 3072)
Volume credits: 2 volumes used (46 credits remain of total 48)
File credits: 4.0 M files reserved (380 M credits remain of total 384
M)

Namespace         Volume Group  Volume              State
---------         ------------  ------              -----
bgh                1      /naumkeag_wing       Enabled

1 Namespace            1 Volume
```

Display information for a specific volume group by specifying a volume group ID as an argument to the command:

```
bstnA# show volume-group 1
```

```
Switch: bstnA
----------------------------------------------------------------------
System Credits
--------------
Share credits: 4 shares used (380 credits remain of total 384)
Direct share credits: 1 direct shares used (6143 credits remain of
total 6144)
Volume credits: 2 volumes used (94 credits remain of total 96)
File credits: 4.0 M files reserved (764 M credits remain of total 768
M)

Volume Group 1
------------
Physical Processor: 1.1
State: Normal; partially used
Share credits: 4 shares used (188 credits remain of total 192)
Direct share credits: 1 direct shares used (3071 credits remain of
total 3072)
Volume credits: 2 volumes used (46 credits remain of total 48)
File credits: 4.0 M files reserved (380 M credits remain of total 384
M)

Namespace         Volume Group  Volume               State
---------         ------------  ------               -----
bgh                1      /naumkeag_wing       Enabled

1 Namespace             1 Volume
```

Display detailed information for a specific volume group by specifying a volume group ID as an argument to the command:

```
bstnA# show volume-group 1 detailed

Switch: bstnA
----------------------------------------------------------------------
System Credits
--------------
Share credits: 4 shares used (380 credits remain of total 384)
Direct share credits: 1 direct shares used (6143 credits remain of
total 6144)
Volume credits: 2 volumes used (94 credits remain of total 96)
File credits: 4.0 M files reserved (764 M credits remain of total 768
M)

Volume Group 1
------------
Physical Processor: 1.1 (29% CPU, 40% MEM)
```

```
State: Normal; partially used
Share credits: 4 shares used (188 credits remain of total 192)
Direct share credits: 1 direct shares used (3071 credits remain of
total 3072)
Volume credits: 2 volumes used (46 credits remain of total 48)
File credits: 4.0 M files reserved (380 M credits remain of total 384
M)

Namespace         Volume Group  Volume             State
---------         ------------  ------             -----
bgh                1      /naumkeag_wing       Enabled


1 Namespace            1 Volume
```

where *volumegroupid* is the integer that identifies the volume group for which you want to display information.

Increasing the Number of Volume Groups Available (ARX-500)

On the ARX-500, the number of volume groups available for use can be increased from the default of two to the maximum supported for that ARX model, using the max-volume-groups CLI command in gbl mode.

The syntax is:

```
bstnA(gbl)# max-volume-groups
```

Executing this command increases the maximum number of volume to groups to the maximum number allowed for the current ARX model.

This command can be negated using the no max-volume-groups command, which returns the number of volume groups to the default of two.

Enabling the Volume

The final step in configuring a direct volume is to enable it. From gbl-ns-vol mode, use the enable command to enable the current volume:

```
enable
```

Direct volumes have no metadata; they only attach to directories on back-end filers. The enable process invokes no imports, so it is much faster than an enable in a managed volume (described in the next chapter). Use show namespace [status] to monitor the progress of the enable: the volume is enabled when all of its shares have a status of “Online: Direct” (or simply “Online” in the show namespace status output).

For example, this command sequence enables the “/vol” volume in the “medco” namespace:

```
bstnA(gbl)# namespace medco volume /vol
bstnA(gbl-ns-vol[medco~/vol])# enable
bstnA(gbl-ns-vol[medco~/vol])# ...
```

Enabling All Shares in the Volume

From gbl-ns-vol mode, you can enable all of the volume’s shares with a single command. Use the enable shares command to do this:

```
enable shares
```

For example, the following command sequence enables all shares in the “medco~/vol” volume:

```
bstnA(gbl)# namespace medco volume /vol
bstnA(gbl-ns-vol[medco~/vol])# enable shares
bstnA(gbl-ns-vol[medco~/vol])# ...
```

Disabling All Shares

Use no enable shares command to disable each of the volume’s individual shares:

```
no enable shares
```

◆ **Important**

This is equivalent to disabling the volume, described below. This causes the volume to stop responding to clients; different client applications react to this in different ways. Some may hang, others may log errors that are invisible to the end user.

For example, this command sequence disables all of the shares in the “access~/G” volume:

```
prtlndA(gbl)# namespace access volume /G
prtlndA(gbl-ns-vol[access~/G])# no enable shares
prtlndA(gbl-ns-vol[access~/G])# ...
```

Disabling the Volume

You can disable a volume to stop clients from accessing it. Use no enable in gbl-ns-vol mode to disable the volume:

```
no enable
```

◆ **Important**

This affects client service. As mentioned above, a disabled volume does not respond to clients; different client applications react to this in different ways. Some may hang, others may log errors that are invisible to the end user.

For example, the following command sequence disables the “medco~/vol” volume:

```
bstnA(gbl)# namespace medco volume /vol
bstnA(gbl-ns-vol[medco~/vol])# no enable
bstnA(gbl-ns-vol[medco~/vol])# ...
```

Showing the Volume

To show only one volume in a namespace, add the volume clause to show namespace command (described in the namespace chapter):

show namespace *name* volume *volume*

where:

name (1-30 characters) is the name of the namespace, and

volume (1-1024 characters) is the path name of the volume.

For a namespace with multiple volumes, the output shows only the volume chosen. The output otherwise matches that of the show namespace command.

For example, the following command shows the configuration of the ‘medco~/vol’ volume:

```
bstnA# show namespace medco volume /vol

Namespace "medco" Configuration
Description: (none)
Metadata Cache Size: 512 MB
NFS Character Encoding: ISO-8859-1

Supported Protocols
-------------------
  nfsv3-tcp

Participating Switches
----------------------
  bstnA (Volume Group 9) [Current Switch]


Volumes
-------
  /vol

             Volume freespace: 116 GB
           Volume total space: 171 GB
                Metadata size: 28 kB
                    Snapshots: Not Enabled
                        State: Enabled

                  Host Switch: bstnA
                     Instance: 1
                 Volume Group: 9
                    Processor: 1.1
                        Files: 1 used (1 dir), 31 M free

    Share corporate
      Filer                       nas1 [192.168.25.21]
      NFS Export                  /vol/vol2
      Status                      Online
      Critical Share              Yes
      Free space on storage       16 GB (17,893,355,520 B)
      Total space on storage      17 GB (18,360,987,648 B)
      Free files on storage       619746
      Virtual inodes              16M
      Transitions                 1
      Last Transition            Wed 07 Nov 2012 01:32:21 AM EST

    Share generic
      Filer                      nas3 [192.168.25.47]
      NFS Export                 /exports
      Status                     Online
      Free space on storage      52 GB (56,814,284,800 B)
      Total space on storage     57 GB (62,229,050,368 B)
      Free files on storage      6M
      Virtual inodes             16M
      Transitions                1
      Last Transition            Wed 07 Nov 2012 01:32:24 AM EST

    Share sales
      Filer                      nas2 [192.168.25.44]
      NFS Export                 /vol/datavol1/direct
      Status                     Online
      Free space on storage      46 GB (50,309,656,576 B)
      Total space on storage     96 GB (103,938,211,840 B)
      Free files on storage      2M
      Virtual inodes             16M
      Transitions                1
      Last Transition            Wed 07 Nov 2012 01:32:23 AM EST


bstnA# ...
```

Showing One Share

To show the configuration and status of one share in a volume, add the share clause after the volume clause:

show namespace *name* volume *volume* share *share-name*

where:

name (1-30 characters) is the name of the namespace,

volume (1-1024 characters) is the path name of the volume, and

share-name (1-64 characters) identifies the share.

This output shows the share that you chose in the command along with its volume and namespace.

For example, the following command shows the configuration of the ‘medco~/vol~corporate’ share:

```
bstnA# show namespace medco volume /vol share corporate

Namespace "medco" Configuration
Description: (none)
Metadata Cache Size: 512 MB
NFS Character Encoding: ISO-8859-1

Supported Protocols
-------------------
  nfsv3-tcp

Participating Switches
----------------------
```

```
  bstnA (Volume Group 9) [Current Switch]


Volumes
-------
  /vol

           Volume freespace: 116 GB
         Volume total space: 171 GB
              Metadata size: 28 kB
                  Snapshots: Not Enabled
                      State: Enabled

                Host Switch: bstnA
                   Instance: 1
               Volume Group: 9
                  Processor: 1.1
                      Files: 1 used (1 dir), 31 M free

    Share corporate
      Filer                       nas1 [192.168.25.21]
      NFS Export                  /vol/vol2
      Status                      Online
      Critical Share              Yes
      Free space on storage       16 GB (17,893,355,520 B)
      Total space on storage      17 GB (18,360,987,648 B)
      Free files on storage       619746
      Virtual inodes              16M
      Transitions                 1
      Last Transition             Wed 07 Nov 2012 01:32:21 AM EST


bstnA# ...
```

Showing Filer Shares Behind One Volume

You can use the show namespace mapping command to show the filer shares behind a particular namespace, as described in the namespace chapter. This shows all attach points in a direct volume and the physical directories behind them. Add the volume clause to show only the shares behind a particular volume:

show namespace mapping *name* volume *volume*

where:

name (1-30 characters) is the name of the namespace, and

volume (1-1024 characters) is the path name of the volume.

The output shows every attach point in the volume, along with the physical directory to which it connects.

For example, this shows the filer shares behind the “medco~/vol” volume:

```
bstnA# show namespace mapping medco volume /vol

Namespace               Physical Server
--------------------    ---------------------
medco:/vol
  vol1/corp             nas1:/vol/vol2/shr
  vol1/notes           nas1:/vol/vol2/notes
  vol2                 nas3:/exports/data
  vol3/mtgMinutes      nas2:/vol/datavol1/direct/mtgs
  vol3/sales           nas2:/vol/datavol1/direct/export


Where * denotes metadata only physical server.
bstnA# ...
```

Showing the Volume’s Configuration

To review the configuration settings for a direct volume, identify the volume at the end of the show global-config namespace command:

show global-config namespace *namespace volume*

where

namespace (1-30 characters) identifies the namespace, and

volume (1-1024 characters) is the volume.

The output shows all of the configuration options required to recreate the volume. The options are in order, so that they can be used as a CLI script.

For example, the following command shows the configuration for the “medco~/vol” volume:

```
bstnA# show global-config namespace medco /vol
;====================== namespace managed volumes =======================
namespace medco
  protocol nfs3tcp
  exit

;====================== namespace direct volumes ========================
namespace medco
  protocol nfs3tcp
  volume /vol
    direct
    share corporate
      critical
      filer nas1 nfs /vol/vol2
      attach vol1/notes to notes
      attach vol1/corp to shr
      enable
      exit

    share generic
      filer nas3 nfs /exports
      attach vol2 to data
      enable
      exit

    share sales
      filer nas2 nfs /vol/datavol1/direct
      attach vol3/sales to export
      attach vol3/mtgMinutes to mtgs
      enable
      exit
    volume-group 9
    enable
    exit

  exit

bstnA# ...
```

Sample - Configuring a Direct Volume

For example, this command set configures the ‘/vol’ volume on the ‘medco’ namespace:

```
bstnA(gbl)# namespace medco volume /vol
bstnA(gbl-ns-vol[medco~/vol])# direct
bstnA(gbl-ns-vol[medco~/vol])# share corporate
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# filer nas1 nfs /vol/vol1/direct
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# attach vol1/corp to shr
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# attach vol1/notes to notes
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# critical
bstnA(gbl-ns-vol-shr[medco~/vol~corporate])# exit
bstnA(gbl-ns-vol[medco~/vol])# share sales
bstnA(gbl-ns-vol-shr[medco~/vol~sales])# filer nas2 nfs /vol/vol1/direct
bstnA(gbl-ns-vol-shr[medco~/vol~sales])# attach vol1/sales to export
bstnA(gbl-ns-vol-shr[medco~/vol~sales])# attach vol1/mtgMinutes to mtgs
bstnA(gbl-ns-vol-shr[medco~/vol~sales])# exit
bstnA(gbl-ns-vol[medco~/vol])# share generic
bstnA(gbl-ns-vol-shr[medco~/vol~generic])# filer nas3 nfs /vol/vol2/direct
bstnA(gbl-ns-vol-shr[medco~/vol~generic])# attach vol2 to data
bstnA(gbl-ns-vol-shr[medco~/vol~generic])# exit
bstnA(gbl-ns-vol[medco~/vol])# volume-group 1
bstnA(gbl-ns-vol[medco~/vol])# enable
bstnA(gbl-ns-vol[medco~/vol])# show namespace status medco

Namespace: medco
Description:

     Share                    Filer                                 Status
         NFS Export
    ------------------------- ------------------------------------- -----------

  Volume: /vol                                                    Enabled
     corporate                nas1                                  Online
         NFS: /vol/vol2

     sales                    nas2                                  Online
         NFS: /vol/datavol1/direct

     generic                  nas3                                  Online
         NFS: /exports

bstnA(gbl-ns-vol[medco~/vol])# exit
bstnA(gbl-ns[medco])# ...
```

Removing a Direct Volume

You can remove a direct volume that is not exported by any front-end service; a later chapter describes how to export or share a volume.

From priv-exec mode, you can use the remove namespace ... volume command to remove a volume:

```
remove namespace name volume volume [timeout seconds] [sync]
```

where:

name (1-30 characters) is the name of the namespace,

volume (1-1024 characters) is the path name of the volume,

seconds (optional, 300-10,000) sets a time limit on each of the removal's component operations, and

sync (optional) waits for the removal to finish before returning. With this option, the CLI lists the volume components as it removes them.

The CLI prompts for confirmation before removing the volume. Enter **yes** to continue.

This operation generates a report, "removeNs_*namespace*_*date*.rpt," which catalogs all of the actions that it took. The *namespace* in the file name identifies the removed namespace, and the *date* is the date and time when the command started. The CLI shows the report name after you invoke the command. Use show reports to see the file listing; use show, tail, or grep to read the file. To save the report off to an external site, use the copy command from priv-exec mode. The command does not create the report if you use the sync option; it shows its actions at the command line instead.

For example, this command sequence removes the '/trialvol' volume from the 'medco' namespace:

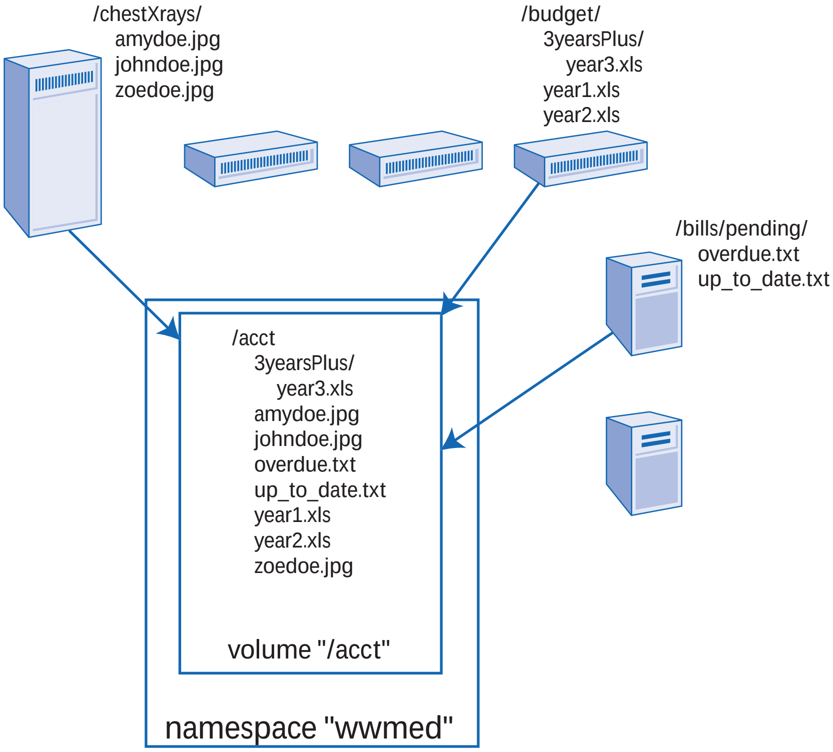
```
bstnA(gbl)# end
bstnA# remove namespace medco volume /trialvol
...
```

9

Adding a Managed Volume

Overview

A *managed volume* aggregates one or more exports/shares from actual filers. The files from each filer share are *imported* into the top directory of the volume. During the share import, the volume catalogues all file and directory locations in its *metadata*. For example, an "/acct" volume with shares from three filers would aggregate files as shown in the figure below:



Metadata facilitates migration policies, but it requires some management. A direct volume, described in the previous chapter, has no metadata and is therefore easier to set up and tear down.

As explained in the namespace chapter, you use the gbl-ns volume command to create a volume (see *Adding a Volume*, on page 7-22). This puts you into gbl-ns-vol mode, where you configure import parameters, metadata storage, at least one share, and several of the options discussed earlier for direct volumes.

For example, this command set creates a single volume ("/acct") for the "wwmed" namespace:

```
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# volume /acct
This will create a new volume.

Create volume '/acct'? [yes/no] yes
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Storing Volume Metadata on a Dedicated Share

A managed volume stores its metadata on a reliable external share. This share, called a *metadata share*, is devoted exclusively to ARX metadata. Best practices dictate that you use a dedicated metadata share for each managed volume. The namespace can have a single metadata share (as described in the *ARX® CLI Reference*) to be divided amongst its managed volumes, but a dedicated share for each volume is preferred.

◆ **Important**

It is vitally important that the external metadata share is fast, highly available, and has multiple gigabytes of free space. A managed volume cannot function if it loses contact with its metadata share.

If the metadata share is an NFS export, we strongly recommend setting the 'sync' and 'no_wdelay' options at the filer export. Other settings slow the performance of the managed volume, especially during import.

If the metadata share resides on a Windows cluster, use the spn *command to inform the ARX of the cluster's virtual SPN. For information about the* spn *command, recall* **Setting the SPN (CIFS)***, on page 6-9.*

An NFS metadata share works for either an NFS or CIFS volume. A CIFS metadata share is not as flexible; an NFS-only volume cannot use a CIFS metadata share because the volume does not have the required proxy-user credentials (see *Choosing a CIFS Proxy User*, on page 7-15) to access a CIFS share.

From gbl-ns-vol mode, use the metadata share command to use a dedicated metadata share for the current volume:

```
metadata share filer {nfs3 | nfs3tcp | cifs} path
```

where

filer (1-64 characters) is the name of the external filer,

nfs3 | nfs3tcp | cifs chooses the protocol to access the share (this can be **nfs3** or **nfs3tcp** for a CIFS-only volume), and

path (1-900 characters for an NFS export, 1-1024 characters for a CIFS share) is the specific export/share on the filer. This external share cannot be used as a namespace share. Use a unique metadata share for each volume. Do not use /vol/vol0 on a NetApp filer; this is reserved for the NetApp operating system, and is not intended for fast access.

cl-name is the name of the ARX cluster with which the metadata share is associated if it is part of a disaster recovery configuration.

For example, this command sequence sets a metadata share to hold all metadata for the "wwmed~/acct" volume:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# metadata share nas1 nfs3 /vol/vol1/meta1
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Removing a Metadata Share

You can remove an unused metadata share from a managed volume.

◆ **Important**

A managed volume requires metadata storage for a successful import. Before you do this, verify that the volume has at least one metadata share before it is enabled.

To remove a metadata share, use the no metadata share command from gbl-ns-vol mode:

```
no metadata share filer {nfs3 | nfs3tcp | cifs} path [cluster cl-name]
```

where

filer (1-64 characters) is the name of the filer,

nfs3 | nfs3tcp | cifs is the file-access protocol, and

path (1-900 characters for NFS, 1-1024 characters for CIFS) is the specific export/share on the filer.

cl-name is the name of the ARX cluster with which the metadata share is associated if it is part of a disaster recovery configuration.

This command has no effect after the volume has already imported its metadata.

For example, this command sequence removes the metadata share on nas3 from the “/rcrds” volume:

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# no metadata share nas3 cifs acp_meta
bstnA(gbl-ns-vol[medarcv~/rcrds])# ...
```

Designating the Metadata Share as Critical (optional)

If the current switch has a redundant peer, you have the option to designate the volume’s metadata share as *critical*. Skip to the next section if this switch is not configured for redundancy.

If the volume software loses contact with its metadata share (that is, if the share’s filer fails to respond to ICMP pings every 30 seconds), the ARX initiates a failover. The failover is accepted by the redundant peer as long as the peer has full access to all critical shares, critical routes, critical metadata shares, *and* the quorum disk. If the peer is unable to access any of these critical resources, no failover occurs. (For instructions on configuring critical routes, refer to *Identifying a Critical Route*, on page 7-18 of the *ARX® CLI Network-Management Guide*.)

If the switch has a redundant peer, we recommend that you use this option for all managed volumes. Without metadata, the managed volume cannot function.

From gbl-ns-vol mode, use the metadata critical command to designate the current volume's metadata share as a critical resource:

```
metadata critical
```

For example, this command sequence designates the metadata share as a critical resource for the "nemed~/acctShdw" volume:

```
prtlndA(gbl)# namespace nemed volume /acctShdw
prtlndA(gbl-ns-vol[nemed~/acctShdw])# metadata critical
prtlndA(gbl-ns-vol[nemed~/acctShdw])# ...
```

Removing Critical-Resource Status

By default, metadata shares are not critical. If the managed volume loses contact with its metadata share in this case, the volume fails and the switch does not initiate a failover. This is not recommended for redundant switches. For non-redundant switches, it is the only option.

From gbl-ns-vol mode, use the no metadata critical command to make the metadata share non-critical for the current volume:

```
no metadata critical
```

For example:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# no metadata critical
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Migrating Metadata to a New Share After Import

After the managed volume is fully enabled, it chooses its metadata share and writes several database files onto it. You may discover that the metadata filer is not as fast or reliable as you would prefer. In this case, you can migrate the volume's metadata to a new filer and share using the Namespace Check (nsck) tool. (The nsck tool and this metadata-migration option is described in the *ARX CLI Maintenance Guide*.)

Allowing the Volume to Modify on Import

When a managed volume imports its shares, it is possible for the shares to have redundant file names (for example, the /etc/hosts file could exist on two different NFS shares). These are called file *collisions*, and they can prevent the import from succeeding. The import can only succeed if you allow the volume to rename the second file and import it with the new name. If such modifications are allowed, the second file is renamed according to the following convention:

filename_share-jobid.ext

where

filename is the file's original name, without its extension (for example, "myfile" from "myfile.doc"),

share is the name of the namespace share (described below),

jobid is an integer identifying the job that imported the share, and

.ext is the file's original extension (for example, ".doc"), if there is one.

If there is more than one redundant file, an index is added:

filename_share-jobid-index.ext

where *index* is a number starting at 1.

Redundant directories are only a problem if their file attributes (such as their permissions settings) do not match, or if they have the same name as an already-imported file. Collided directories are renamed according to the same convention as files.

◆ **Note**

For a multi-protocol (NFS and CIFS) namespace, directories are also renamed if their CIFS names are not mappable to the NFS-character encoding. This was discussed in Setting Up Character Encoding, on page 7-11.

You will have the option to disallow file renames and/or directory renames for individual shares. These are discussed later in the share-configuration section.

From gbl-ns-vol mode, use the modify command to allow the volume to modify redundant files and/or directories on import:

```
modify
```

Clients cannot write to the volume until you run the modify command.

The CLI prompts with a question about re-imports. The *ARX CLI Maintenance Guide* describes how to use a namespace-check tool (nsck) to re-import a previously-enabled volume. This tool turns off the modify setting and takes the volume offline. The modify setting stays off during re-import unless you raise a flag by answering this question. Answer "yes" to allow the modify setting to remain if the volume is rebuilt through nsck. A "no" makes the volume read-only after any re-import.

For example, this command sequence allows the '/acct' volume to modify files on import and re-import.

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# modify
Automatically re-enable volume modify mode during nsck rebuild? [yes/no] yes
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Conditions for Running the modify Command

You can run the modify command

- before the volume is first enabled, or
- after an import with no modify (assuming no file or directory collisions occurred on import).

You cannot use the modify command if the volume is in the process of importing, if any imported shares have collisions, or if the nsck utility is being used on the volume.

Running a No-Modify Import

If you choose to import without using the modify command (to run a hypothetical import) and there are no file collisions, you can later raise the modify flag. This completes the import and gives write access to clients.

You can use the shares' import reports to find any collisions that would have occurred. Import reports appear after the managed volume is enabled. Use show reports type Imp to find all import reports on the system; import reports are named "import.*share-id*.*share-name*.*job-id*.rpt." Use show reports *report-name* to view the report. You can correct any collisions directly on the back-end filer(s), remove the share(s) from the volume (as described in *Removing an Imported Share*, on page 10-18 of the *ARX CLI Maintenance Guide*), and add each share back into the volume to re-import it.

Allowing the Volume to Modify on Re-Import

As mentioned above, the *ARX CLI Maintenance Guide* describes the nsck tool for rebuilding a managed volume. If you said "no" to the CLI prompt after you used the modify command, nsck does not raise the modify flag after taking the volume offline. This means that the volume will be read-only after a volume rebuild. You can use another CLI command, reimport-modify, to allow the modify flag to stay up after the volume is rebuilt:

reimport-modify

This command prompts for confirmation before taking action; enter **yes** to proceed.

For example, this command sequence ensures that the "/acct" volume will be modifiable after any re import:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# reimport-modify

Automatically re-enable volume modify mode during NSCK rebuild? [yes/no] yes
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Preventing Modification On or After Re-Import

Use the no reimport-modify command to keep the modify flag down after using nsck:

```
no reimport-modify
```

This is the default.

For example:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# no reimport-modify
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Preventing Modifications

By default, the managed volume is read-only after its initial import, and it does not modify any back-end files during the import. Use no modify command to return to this default:

```
no modify
```

You can only run this command on a volume with no imported shares. That is, you can run this before you first enable the volume, or after taking the volume offline with the nsck tool.

For example:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# no modify
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Automatically Synchronizing Metadata (CIFS)

This section only applies to volumes in namespaces that support CIFS. Skip to the next section if the namespace is NFS-only.

If a file becomes missing on a filer without the managed volume's knowledge, the volume's metadata is compromised. This can happen if one of the filer's local applications, such as anti-virus software, deletes a file or moves it to a quarantine area. If a client tries to access the file through the volume, the client gets an error. You can configure the managed volume to automatically detect this error and synchronize its metadata with the actual files. From gbl-ns-vol mode, use the auto sync files command to allow the volume to automatically synchronize its metadata:

```
auto sync files
```

For example, the following command sequence allows the "medarcv~/rcrds" volume to automatically sync its metadata:

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# auto sync files
bstnA(gbl-ns-vol[medarcv~/rcrds])# ...
```

◆ **Note**

A file that is added on a backend filer without the ARX's knowledge will be recognized by a managed volume only after a manual sync files *operation is performed.*

Auto-Sync Jobs and Reports

The volume launches a sync job whenever a filer returns an error indicating that the volume's metadata is wrong. Each auto-sync job gets a unique job ID, similar to the job IDs for import jobs. It also generates a report; use the show reports type sync command to list all sync reports, or use show reports *report-name* to examine a report's contents. All auto-sync reports are named as follows:

auto-sync.*job-id*.*vol-path*.rpt

where

job-id is an integer identifying the job that synchronized the volume, and

vol-path is the volume's name, where all slashes (/) are replaced with underscores (_).

Allowing Renames on Collision

An auto-sync job may discover a file that collides with another file in the volume (that is, in another share). By default, this prevents the operation from synchronizing that file; a managed volume cannot support two or more files whose path names collide. To work around these collisions, you can configure the volume to rename these files before using them. The volume uses the following pattern to rename the file:

filename_share-jobid.ext

where

filename is the file's original name, without its extension (for example, "zapFiles" from "zapFiles.exe"),

share is the name of the volume share (described below),

jobid is an integer identifying the job that synchronized the volume, and

.ext is the file's original extension (for example, ".exe"), if there is one.

If there is more than one redundant file, an index is added:

filename_share-jobid-index.ext

where *index* is a number starting at 1.

This renaming convention is the same one that is used for file collisions on import (recall *Allowing the Volume to Modify on Import*, on page 9-6).

From gbl-ns-vol mode, use the rename-files flag at the end of the auto sync files command to allow the volume to rename any collision files:

```
auto sync files rename-files
```

For example, the following command sequence allows sync-file renames in the "medarcv~/rcrds" volume:

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# auto sync files rename-files
bstnA(gbl-ns-vol[medarcv~/rcrds])# ...
```

Disallowing Renames

If auto-sync jobs are not allowed to rename files that collide, those files cannot be synchronized. The metadata for those files remains stale, so clients cannot access them. To disallow renames, use the no auto sync files rename-files command:

```
no auto sync files rename-files
```

For example,

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# no auto sync files rename-files
bstnA(gbl-ns-vol[medarcv~/rcrds])# ...
```

Disabling File Auto-Synchronization

The *ARX CLI Maintenance Guide* describes a manual *sync* utility that you can use to synchronize metadata on command. You can use this utility as needed. From gbl-ns-vol mode, use no auto sync files to stop the automatic syncs and rely exclusively on the manual utility:

```
no auto sync files
```

For example:

```
bstnA(gbl)# namespace medarcv volume /itvol
bstnA(gbl-ns-vol[medarcv~/itvol])# no auto sync files
bstnA(gbl-ns-vol[medarcv~/itvol])# ...
```

Configuring the Volume’s Free Space Calculation (Optional)

The next step in creating a volume is to choose an algorithm for calculating its free space. This is the free space calculation that is passed on to the client: whenever a user mounts a volume (NFS) or maps a network drive to it (CIFS), this total is the free space that they see.

By default, the namespace volume’s free space is the sum of the free space on all of its back-end shares *except* shares from the same storage volume. If two or more shares report the same ID for their backing volume, only the first share is counted. (For NFS shares, the volume uses the file system ID (FSID); for CIFS shares, it uses the Volume Serial Number). This is known as automatic free space calculation. If this default is acceptable, you can skip the rest of this section.

Free space calculation is configured using the freespace calculation CLI command. The command’s syntax is:

```
freespace calculation {manual | dir-master-only}
no freespace calculation
```

◆ **Note**

The default configuration is no freespace calculation*, causing automatic free space calculation to be performed.*

The alternatives to automatic free space calculation that can be enabled by this command, manual free space calculation and directory master free space calculation are described in the sections that follow.

Using Manual Free Space Calculation

You may wish to control the free space calculation manually on a share-by-share basis. This means counting all free space on all shares, regardless of duplicate back-end-volume IDs, then ignoring certain shares manually or adjusting their free space reporting. You can use the freespace ignore and freespace adjust commands, described later, to ignore the free space from a share or change the reported value for the free space.

From gbl-ns-vol mode, use the freespace calculation manual command to override the default free space calculation:

```
freespace calculation manual
```

You can set this at any time, even after the volume is enabled.

For example, this command sequence makes the ‘access~/G’ volume count the free space in all back-end shares, even multiple shares that draw from the same back-end storage:

```
prtlndA(gbl)# namespace access volume /G
prtlndA(gbl-ns-vol[access~/G])# freespace calculation manual
prtlndA(gbl-ns-vol[access~/G])# ...
```

Ignoring a Share's Free Space (optional)

This option is only relevant in a volume where you are manually calculating free space (see *Configuring the Volume's Free Space Calculation (Optional)*, on page 9-13). Such a volume's free space is the sum of the space from *all* of its shares, including multiple shares from the same back-end storage volume. This can mean counting the same storage multiple times: two or more shares from the same storage volume each report the full amount of free space on the volume. For example, two NFS shares from the same disk partition, /lhome, would each report the total free space on the /lhome partition. A volume with both of these shares would double-count the free space in /lhome.

You can manually exclude shares from the free space calculation using freespace ignore, as described for direct shares. For example, this command sequence ignores all free space in the "back2" share:

```
prtlndA(gbl)# namespace nemed volume /acctShdw share back2
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back2])# freespace ignore
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back2])# ...
```

Including a Share in the Free Space Calculation

By default, free space from all shares is counted toward the volume's total. To include a share in the free space calculation, use no freespace ignore as you would with a direct share (recall *Including the Share in the Free Space Calculation*, on page 8-15). For example:

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# no freespace ignore
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Adjusting the Free Space Calculation

You can also manually adjust the free space that is advertised for a share. From gbl-ns-vol-share mode, use the freespace adjust command. This was described in detail for direct volumes; see *Adjusting the Free Space Calculation*, on page 8-16. For example, this command sequence adds 1 gigabyte to the "back1" share's free space calculation:

```
prtlndA(gbl)# namespace nemed volume /acctShdw share back1
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back1])# freespace adjust 1G
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back2])# ...
```

Removing the Free Space Adjustment

Use the no freespace adjust command to remove any free space adjustment from a share. For example, this command sequence removes any free space adjustment from the "back2" share:

```
prtlndA(gbl)# namespace nemed volume /acctShdw share back2
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back2])# no freespace adjust
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back2])# ...
```

Using Directory Master Free Space Calculation

It is possible also to display the free space only for the back-end file system that is being accessed, rather than for the entire managed volume. This is useful in cases such as file migration, in which the temporary existence of two filesystems for the one in migration could otherwise distort the results of volume size and free space reporting. This is accomplished via the CLI command, freespace calculation dir-master-only, which causes the share in question to be queried only on the storage resource at which its master instance resides. The syntax for this command is:

```
freespace calculation dir-master-only
```

In addition, the freespace apparent-size CLI command in gbl-ns-vol-shr mode enables you to configure an artificial capacity value for the shares of the volume that is different than its actual capacity. This command is relevant only when free space is calculated using the directory master only algorithm.

The syntax for the freespace apparent-size command is:

```
freespace apparent-size sharevalue
```

where *sharevalue* is the amount of artificial capacity to configure for the corresponding share. This value can be followed by k, M, G, or T to indicate the corresponding units.

This functionality has no effect on policies and shadow volumes, for which actual free space, not apparent size and free space, is in use at all times.

Using Automatic Free Space Calculation (Default)

By default, free space is calculated based on the IDs of the volumes behind the back-end shares. If any of these shares report the same volume ID, their free space is counted only once. To return to this default, use the no freespace calculation command:

```
no freespace calculation
```

For example:

```
bstnA(gbl)# namespace medco volume /vol
bstnA(gbl-ns-vol[medco~/vol])# no freespace calculation
bstnA(gbl-ns-vol[medco~/vol])# ...
```

Free Space Reporting For Path-Based Quotas

It is possible to display the free space for a volume in the context of the user account and path used to access the volume, a practice supporting path-based quotas. The freespace cifs-quota CLI command in gbl-ns-vol mode enables volume free space to be displayed based on the credentials of the user executing the command and the path by which the volume is accessed.

This behavior is disabled by default (no freespace cifs-quota), causing the system-wide free space algorithm to be used. The freespace ignore and freespace adjust commands can be used in conjunction with this as well.

CIFS clients see the results of this command only if they connect after you invoke it.

◆ **Note**

This command pertains specifically to CIFS clients, and has no effect upon NFS queries.

◆ **Note**

The commands freespace calculation dir-master-only *and* freespace cifs-quota *are mutually exclusive. In the event that both commands were executed,* freespace cifs-quota *would take precedence for CIFS clients, and* freespace calculation dir-master-only *would take precedence for NFS clients.*

NetApp and EMC With Filer Subshares

Free space quotas (this command) and filer subshares are difficult to use together in a volume backed by either NetApp or EMC. A NetApp only supports path-based quotas on its qtrees, and an EMC server only supports them on its filesystems. Each qtree or filesystem gets a single quota. As mentioned above, each subshare requires a path-based quota for proper free space reporting. Therefore, each subshare path must be backed by one qtree or filesystem. This is required for CIFS clients to see the space that they are allowed to use.

The ARX volume automatically replicates its subshare directories to all of its back-end filers. The replication process cannot safely presume that you want a full qtree or filesystem behind these replicated directories, so it can only create the directory itself. Therefore, the subshare replication process creates directories on NetApp and EMC that cannot support path-based quotas.

For NetApp and EMC, pre-create the qtrees and/or filesystems instead of relying on the ARX volume's replication process. Contact F5 if you need assistance with this configuration.

Relationship of Front-End Shares to Back-End Quotas

Path-based quotas are applied to back-end paths, so the free space that the client sees depends on the front-end CIFS share to which it connects. There are two major forms of front-end CIFS share:

- An export (gbl-cifs) is a full share of the entire volume, and
- An export (gbl-cifs) ... filer-subshare is a subshare.

If a CIFS client connects to a full share (or any path within a full share), the client sees the free space quotas set at the imported path (or root) of the volume's back-end shares. If the client connects to a subshare, the client sees the free space quotas set at its back-end subshare paths.

If each subshare is dedicated to a single client (as with a home directory application), you can apply path-based quotas at each back-end subshare and then use the freespace cifs-quota command to advertise those quotas to the volume's clients.

◆ **Note**

We recommend setting specific path-based quotas at each back-end share or subshare to which your CIFS clients connect. For NetApp and EMC, this means that each share or subshare must be its own qtree (for NetApp) or filesystem (for EMC).

Back-End Shares That Draw From the Same Storage Pool

An ARX volume may import two or more shares that are affected by the same free space number, and this creates errors in free space reporting. Whenever the freespace cifs-quota feature is enabled, the ARX volume presumes that a path-based quota is set on all of its back-end shares; it therefore counts the space from each share separately. For example, suppose shares HUEY, DEWEY, and LOUIE all draw from drive E on the same Windows server, and an ARX volume imports all three of them. The volume also adds together all of their free space. This creates an incorrect sum if:

- there is no quota (all three shares report the full space of the E drive),
- there is a user-based quota (all shares report the full space allowed to the client),

or

- all three shares inherit the same path-based quota from a common ancestor, such as E:\.

Specific path-based quotas for each share, recommended earlier, helps to solve this problem. Additionally, avoid any user-based quotas or common path-based quotas altogether.

If this is not possible, use freespace ignore to ignore each imported share from a common storage pool except one. For example, ignore the freespace on any two of the three shares (such as HUEY and DEWEY).

Setting CIFS Options

The next step in configuring a volume is addressing its CIFS options, if necessary. Skip to the next section if this volume is in an NFS-only namespace.

There are five CIFS-volume attributes that back-end filers may or may not support. They are named streams, compressed files, persistent ACLs, Unicode file names on disk, and sparse files. Each volume can support any and all of these capabilities. However, *all* filer shares used in a volume *must* support the capabilities advertised in the volume. For example: if your volume is going to support compressed files, then *all* of its back-end filer shares must also support compressed files. The show exports command displays the CIFS options on back-end filers, as described in *Showing CIFS Attributes*, on page 5-13.

The CIFS options conform to those of the first-enabled share by default; the options are scanned during the share import. To manually control these options, you can use the same commands that you use in a direct volume (recall *Setting CIFS Options*, on page 8-6). That is, you can use any combination of the following gbl-ns-vol commands to manually set the options:

```
[no] named-streams
[no] compressed-files
[no] persistent-acls
[no] unicode-on-disk
[no] sparse-files
```

For example, the following command sequence disables two CIFS options in the “medarcv~/test” volume:

```
bstnA(gbl)# namespace medarcv volume /test
bstnA(gbl-ns-vol[medarcv~/test])# no named-streams
bstnA(gbl-ns-vol[medarcv~/test])# no sparse-files
bstnA(gbl-ns-vol[medarcv~/test])# ...
```

CIFS clients see the results of the named-streams command only if they connect after you invoke it.

File-System Type Advertised by the Volume

CIFS-Client applications have the option to request the managed volume's file-system type. The managed volume uses its CIFS options to determine its answer for these queries:

File-System Type Advertised	Volume's CIFS Options
FAT	(all disabled)
FAT32	unicode-on-disk
NTFS	unicode-on-disk persistent-acls named-streams

◆ **Note**

The volume also advertises itself as a FAT file system if any of its back-end shares do not support "case preservation." This is a feature supported by any file system that supports long names (names longer than the old "8.3" format), so it is nearly ubiquitous.

The volume can only support the CIFS options supported by *all* of its back-end shares. For example, a volume that is backed by three Windows XP shares and two Samba shares can only use the lesser CIFS options of the Samba shares.

Client applications use this information to interpret various data from the managed volume, such as file-access times. If any of the volume's shares is backed by a FAT file system, the volume's clients only see old-style access times without specific hours, minutes, or seconds. This applies to the entire volume, not just the share with the FAT file system.

Supporting Filers with Local Groups

A Windows filer can support *Global Groups*, which are managed by domain controllers, and/or *Local Groups*, which are unique to the filer. Local groups have their own Security IDs (SIDs), unknown to any other Windows machine. When you aggregate shares from these filers into a single volume, some files tagged for local-group X are likely to migrate to another filer, which does not recognize the SID for that group (SID X). This invalidates any Access Control Entries (ACEs) with SID X while the file resides on that filer; members of group X lose their privileges. To resolve this problem, you must first prepare all of the filers before you aggregate them into a namespace volume:

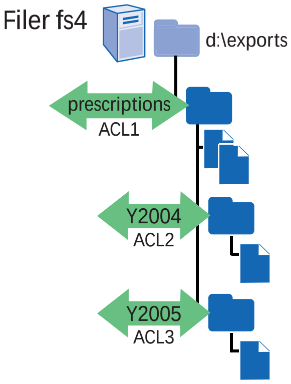
- all local-group names must be configured on all filers behind the volume, and

- all groups must contain the same users on those filers.

For example, if filer A has local group “doctors” and filer B has local group “nurses,” you must add a new “nurses” local group to filer A and a new “doctors” group to filer B. The membership of these groups must match at both filers. This preparation is required so that all doctors and nurses can share the filers behind a volume.

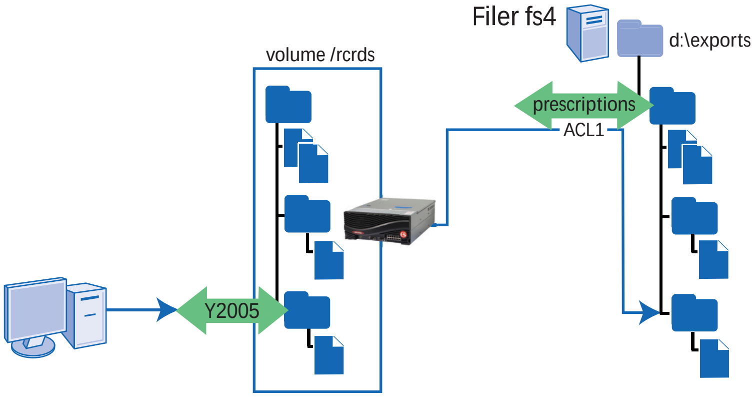
Supporting Subshares and their ACLs

Windows filers can share multiple directories in the same tree, and can apply a different share-level Access Control List (ACL) to each of them. Consider the following three shares on the fs4 filer:



A client who connects to the “prescriptions” share has the access rights defined in ACL1, whereas the same client uses ACL3 if he or she connects to the “Y2005” *subshare*. ACL1 may restrict the client to read-only access, while ACL3 gives the gives the client full control.

By default, all managed-volume access comes through the top-level share at the back-end filer (“prescriptions” in this example). The volume’s CIFS clients would therefore always have the permissions defined by ACL1, even if they connected through the front-end equivalent of the filer’s Y2005 subshare:



To prepare the managed volume to pass connections through to the filer's subshares, thereby using the subshares' ACLs, use the gbl-ns-vol filer-subshares command:

```
filer-subshares
```

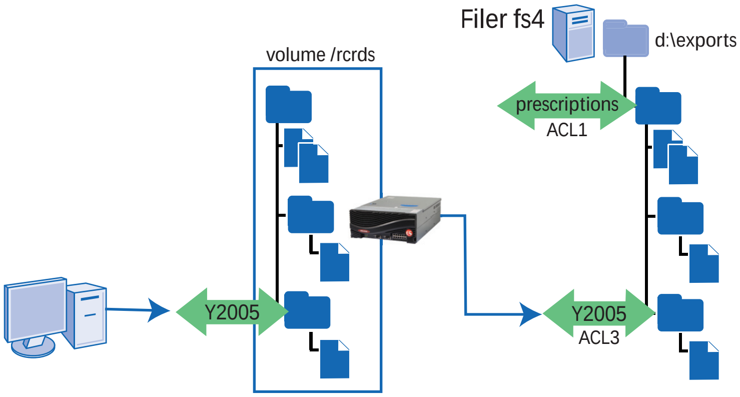
You cannot use this command while any of the volume's shares are enabled.

This command only prepares the volume for subshare support at the back-end. *Chapter 11, Configuring Front-End Services* describes how to configure the client-visible subshares in a front-end CIFS service, like the front-end "Y2005" share in these illustrations.

For example, the following command sequence prepares the "/rcrds" volume to support filer subshares and their back-end ACLs:

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# filer-subshares
bstnA(gbl-ns-vol[medarcv~/rcrds])# ...
```

This causes the managed volume to recognize subshares. When you later configure a "Y2005" subshare in the front-end CIFS service, clients who connect to the subshare will use the correct share-level ACLs:



Using the filer-subshares command with its optional native-names-only argument enables subshare support without also allowing the renaming of conflicting share names with generated names. When the native-names-only argument is used, import operations will fail if any shares need a generated name to resolve a naming conflict.

◆ **Note**

If subshares are to be used, all shares must be configured with the import sync-attributes *command, which causes directories to be renamed if they encounter naming conflicts.*

For NetApp filers and EMC servers, manually replicate the subshares as qtrees/filesystems before you enable the volume. Do this at every NetApp and EMC share behind this managed volume. Use the same directory name, relative path, and share name for all of them.

To add a new subshare in a running volume that already supports them, choose an existing directory to be exported as a subshare. If none exists, create it.

If you want to support CIFS quotas (freespace cifs-quota) and the volume is backed by an EMC server or a NetApp filer, go directly to the back-end filer(s) and manually create filesystem(s) or qtree(s) for the subshare. Use the same directory name and relative path for all of them. Then run sync directories to incorporate the new directory into the volume metadata.

If you are not supporting quotas from NetApp or EMC, connect to the front-end export (gbl-cifs) as a client and create its directory.

Required Windows Permissions

To read the shares' paths at the back-end filers, the volume requires proxy-user credentials with Administrator-level access. This is a significant increase in access from the standard proxy-user requirements; you may need to increase the permissions for the proxy user on all filers, or use a more-powerful proxy user for this namespace. For instructions on editing a proxy user, recall *Adding a Proxy User*, on page 3-4. *Choosing a CIFS Proxy User*, on page 7-15 describes how to add new proxy-user credentials to the namespace.

Showing a Filer's Shares and Subshares

You can use the show exports ... path command to display the physical paths for all CIFS shares on a back-end filer. This was discussed in an earlier chapter: recall *Showing the Physical Paths for CIFS Shares*, on page 5-8. The output from this command shows all of the share-to-subshare relationships on the filer. For example, this command shows one share ("prescriptions") and its subshares ("CELEBS$," "Y2004," "Y2005," and "Y2010") on the "fs4" filer:

```
bstnA(gbl)# show exports external-filer fs4 paths proxy-user acoProxy2
Export probe of filer "fs4" at 192.168.25.29

CIFS Credentials:
    User              jqpublic
    Windows Domain    MEDARCH.ORG
    Pre-Win2k Domain  MEDARCH

Paths:

  CIFS

    Share                            Directory
    ---------------------------      ------------------------------------
    CELEBS$                          e:\exports\prescriptions\VIP_wing
    Y2004                            e:\exports\prescriptions\2004
    Y2005                            e:\exports\prescriptions\2005
    Y2010                            e:\exports\prescriptions\2010
    prescriptions                    e:\exports\prescriptions
    ---------------------------      ------------------------------------
    Share                            Directory

bstnA(gbl)# ...
```

Disabling Filer Subshares

From gbl-ns-vol mode, use the no filer-subshares command to disable volume support for CIFS subshares and their share-level ACLs:

```
no filer-subshares
```

You can only disable this feature when no CIFS services are sharing any of the volume’s subshares. CIFS front-end services are described in a later chapter, along with instructions on sharing CIFS subshares.

For example, this command sequence prevents CIFS-subshare support in the “insur~/claims” volume:

```
bstnA(gbl)# namespace insur volume /claims
bstnA(gbl-ns-vol[insur~/claims])# no filer-subshares
bstnA(gbl-ns-vol[insur~/claims])# ...
```

Replicating Subshares at all of the Volume’s Filers

The managed volume must have consistent subshares and subshare ACLs on all its back-end shares. Consistency is required so that clients have the same permissions no matter which back-end share contains their files and directories.

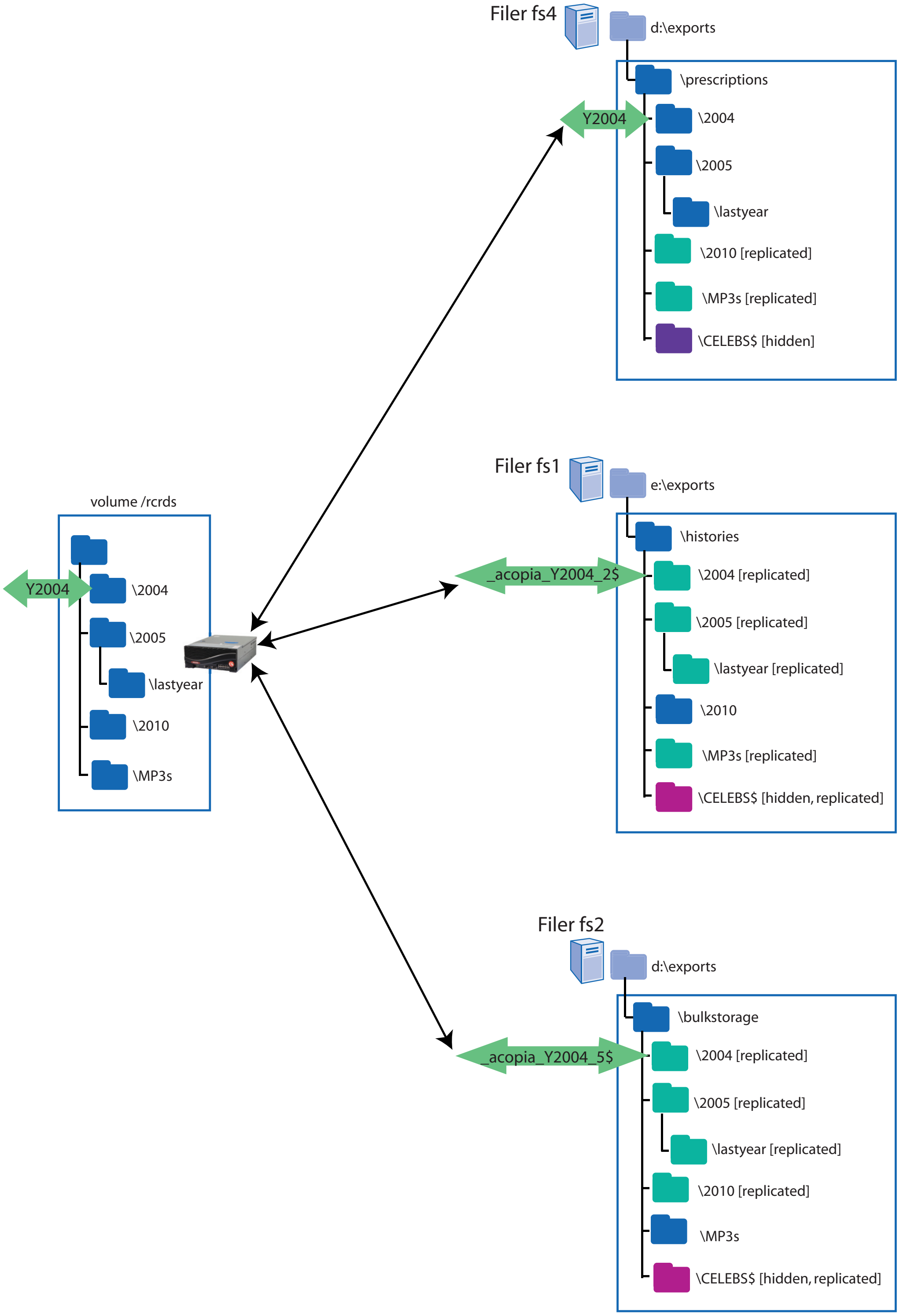
As of the current release, replication occurs automatically when subshare functionality is enabled using the filer-subshares command. The volume copies all subshare definitions from each share to all the other shares. If necessary, the volume copies underlying directories as well. Subshares are replicated to newly-added shares as well.

The ARX attempts to replicate using native subshare names whenever possible. If there is a naming conflict and it is necessary to use a generated name for a replicated subshare, the generated name follows this convention:

> _acopia_*subshare*_*id*$
>
> where
>
> > *subshare* is the name of the original subshare (such as “Y2004”).
> >
> > *id* is an integer that uniquely identifies the subshare. This ID is different for every CIFS subshare behind the ARX and its redundant peer.
> >
> > $, at the end of the subshare name, hides the subshare from a standard **net view** or similar operation.

This filer subshare is behind the ARX volume, and should not be accessible to CIFS clients. The subshare name is meant to be usable by ARX software, potentially on behalf of CIFS clients.

When each share is enabled (as described later), the volume reads all of its subshare information and replicates it to the remaining shares.



Synchronizing Subshares

Subshare synchronization ensures the consistency of back-end subshares that correspond to front-end CIFS service exports. It identifies all of the subshares that are associated with a managed volume, and ensures that each subshare is replicated on each filer associated with the volume. This N-way synchronization of a volume's relevant backend shares, where N is the number of filers hosting the backend shares, is useful for repairing broken subshare stripes.

Subshare synchronization can be performed from a managed volume to a service or from a service to a managed volume, using the privileged-exec mode command sync subshares. The managed volume must be in modify mode in order for subshare synchronization to be performed.

◆ **Note**

The managed volume in question must be configured using the filer-subshares command.

◆ **Important**

If the volume is backed by any NetApp filers or EMC servers and you plan to support free-space quotas on them (using freespace cifs-quota), prepare the subshares directly on the filer before you synchronize subshares. The subshare-replication process does not create NetApp qtrees or EMC filesystems, and those special directories are required to support free-space quotas. Before you run subshare replication, access the filers directly and create one qtree or filesystem per subshare. Then, synchronize their share-level ACLs and other attributes.

Synchronizing from a volume to a service

This privileged-exec mode command sync subshares from-namespace replicates subshares and their corresponding ACLs between filers as needed.

The operation discovers all of the backend subshares behind a managed volume, synchronizes them (N-way across the relevant filers) so that all import shares have all subshares, and creates exports in the target service for that set of subshares.

The command's syntax is as follows:

```
sync subshares from-namespace ns volume vol to-service fqdn
[expose-hidden] [tentative]
```

If the expose-hidden option is used, consistent backend subshares for which the native name ends in “$” will be exported via the service with the “$” stripped away. If both a “$” and a non-”$” version of a backend subshare exist, the ARX will report this fact and use the “$” subshare only if the expose-hidden option is used.

◆ **Note**

This command replaces the cifs export-subshares *command that was available in earlier releases.*

◆ **Note**

If the managed volume is not in modify mode, and the volume or the back-end needs to be modified to be made consistent, you must execute the command with the tentative *option. This provides a report but does not attempt to execute the actions necessary to synchronize the subshares.*

Synchronizing from a service to a volume

The privileged-exec mode command sync subshares from-service replicates subshares and their corresponding ACLs between filers as needed by the front-end exports of the specified service. Synchronizing from a service to a volume means using the service’s existing file-subshare exports to select which back-end subshares in the volume are to be synchronized, leaving other backend subshares alone.

The operation examines subshare exports in the specified service which point into the specified volume and, as necessary, creates or updates the corresponding back-end subshare stripes that match the front-end exports. Back-end subshares not matching any front-end export are left undisturbed.

The command’s syntax is as follows:

```
sync subshares from-service fqdn to-namespace ns volume vol
[tentative]
```

◆ **Note**

If the managed volume is not in modify mode, and the volume or the back-end needs to be modified to be made consistent, you must execute the command with the tentative *option. This provides a report but does not attempt to execute the actions necessary to synchronize the subshares.*

Promoting Subshares

Subshare promotion is an operation for changing the name of a back-end subshare from a generated name (generated by the ARX to avoid a naming conflict) to a native name. This is useful for changing the names of back-end subshares that may have acquired generated names from earlier releases of the ARX software. Use the cifs promote-subshares command to do this.

Execute the privileged-exec mode command cifs promote-subshares to promote a subshare. This command can be executed per import share, and promotes all of the subshares under the specified import share.

◆ **Note**

Subshare promotion is an emergency recovery operation. There is no need to execute the cifs promote-subshares *command if subshares using ARX-generated names are working properly.*

The Subshare Synch/Promote Report

Subshare promotion and synchronization operations create reports that detail the actions that are executed or, in tentative mode, that will be executed. If the sync subshares or cifs promote-subshares command is issued with the tentative option, the report is generated, but none of the actions are actually executed. In this case, the report provides a preview of the actions that would be executed to make the affected subshares operational.

Display the report using the command show reports *reportname*, where *reportname* is the name assigned to the report by the system upon execution of the sync subshares command or cifs promote-subshares command.

The report includes the following information:

- operation parameters
- share information
- options
- operation execution detail
- summary

Supporting Access-Based-Enumeration (ABE)

Some Windows servers offer a security feature called *Access-Based Enumeration* (ABE). An ABE-enabled share customizes its directory listings for each CIFS client; the list only contains the files and folders where the client has read access. The intent of this feature is to reduce client curiosity about files and directories that they are prohibited from reading.

ABE-enabled shares cause a managed volume to mistakenly assume that files are missing from the back-end filer. The volume monitors all of its clients' file listings as they occur, and compares those listings to its metadata. The volume uses the client's identity when it asks the back-end share for a directory listing, so the listing may be incomplete if the share has ABE enabled. For example, suppose Client A creates a file, "classified.doc," and sets its permissions so that Client B cannot read it. The ABE-enabled share includes the file in listings for Client A, but omits it from listings for

Client B. If Client B asks for a listing of the directory, the volume notes that “classified.doc” is missing. If Client A asks for the same directory listing later, the volume then notes that the “classified.doc” file has reappeared.

By default, a volume adds missing files to a file listing before it passes the list back to the client. The volume assumes a metadata inconsistency, possibly caused by a client error or anti-virus software on the filer, and reports the last-known list of files and subdirectories. This feature causes a problem in a volume backed by ABE-enabled shares: it exposes inaccessible files to clients who are not allowed to view them. To continue the above example, the volume would show “classified.doc” to Client B.

From gbl-ns-vol mode, use the cifs access-based-enum command to inform the volume that its shares use ABE. Subsequent to this, shares will be checked for ABE support at the time that they are first imported to the volume, and the volume will stop amending sparse file listings for its clients. The command syntax is:

```
cifs access-based-enum
```

This command indicates also that you want ABE consistently enabled for all shares and subshares behind the volume. If you already have shares imported *without* ABE enabled, a prompt asks you to accept this inconsistency; enter **yes** to continue. (There is a command that you can use to enable ABE on all of the volume’s back-end shares, described later in this section.)

◆ **Note**

As of Software Release 5.02, the ARX checks a share’s ABE status only at the time that the share is first imported. This is a departure from its behavior in releases prior to Release 5.02, in which it would check ABE status each time that a share was enabled or disabled. The ABE check is performed in the background, and no explicit notification is given if the ABE configuration is inconsistent. Use the show namespace status *CLI command to determine whether a share import was successful, or failed due to ABE inconsistency. If an existing share is enabled and its back-end ABE status is inconsistent with its status on the volume, the back-end share can be forced into agreement using the* cifs access-based-enum <*namespace*> <*volume*> *CLI command described in Adding ABE to a Running Volume, on page 9-31.*

For example, the following command sequence supports ABE in the “medarcv~/lab_equipment” volume:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# cifs access-based-enum
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# ...
```

After you set cifs access-based-enum, the volume checks new shares for ABE consistency when they are first imported. If ABE is disabled at the filer, the volume does not allow you to enable the share in this managed volume.

ABE Requires Persistent ACLs

Some filers cannot consistently support CIFS ACLs at all, and therefore preclude any ABE support in the volume. If the volume is backed with any of these filers, you must set no persistent-acls at the volume before it can import from any such filers (recall *Setting CIFS Options*, on page 9-18). The volume therefore ignores ACLs entirely, and does not preserve them when it migrates files between back-end shares. A volume with this setting cannot support ABE, because ABE depends on the permissions settings in file ACLs.

Consistent ABE Support Behind the Volume

A managed volume can aggregate storage from multiple shares, and it is possible for those back-end shares to have inconsistent ABE settings. This inconsistency may confuse the volume's CIFS clients. For example, if "\myDocs\secretFile.doc" migrates from share X (where ABE is enabled) to share Y (where ABE is disabled), some ARX clients see it suddenly appear in the "\myDocs" directory. If the file migrates back to share X, it suddenly disappears from the clients' directory listings. To prevent this confusion, the volume does not allow you to enable any new shares where ABE is disabled on the backing filer.

You have the option to automatically enable ABE on the back-end share as each managed-volume share is imported. This makes it possible to incorporate new shares into the volume while ensuring ABE consistency. From gbl-ns-vol mode, add the auto-enable flag to the end of the cifs access-based-enum command:

```
cifs access-based-enum auto-enable
```

where **auto-enable** (optional) causes the volume to enable ABE on any of the volume's back-end shares at the time they are imported.

◆ **Important**

If a share's backing filer cannot support ABE, the auto-enable *option will fail for the share. ABE support has been tested with filer releases as early as Windows 2003 R1, NetApp 7.2.3, and EMC 5.5. It is not supported by earlier releases, or by any release of Samba.*

To allow the addition of an ABE-incapable share, there is an option to exclude the share from ABE support. This exclude option is described later, in the section about shares.

The auto-enable option has no effect on currently-enabled shares in the volume.

For example, the following command sequence enables ABE for the "medarcv~/lab_equipment" volume, and ensures that the volume will enable ABE at the back-end filer behind any new share that is imported:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# cifs access-based-enum auto-enable
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# ...
```

Windows Server Clusters and ABE

Some Windows servers support a redundancy feature, called Windows Server Clustering, that complicates ABE support. The auto-enable flag only enables ABE on the currently-active node in a Windows Server Cluster. If the cluster fails over, the newly-active node may have different ABE settings. This is likely to introduce the ABE inconsistency described previously. To avoid this issue, manually set the ABE state in the cluster's administrative interface, so that the cluster enables ABE on all of its nodes. If the cluster's interface does not offer an option for setting ABE, enable it on all of the cluster's nodes manually.

Disabling ABE

From gbl-ns-vol mode, use no cifs access-based-enum to disable ABE at the volume level:

```
no cifs access-based-enum
```

This indicates that all directory listings should be consistent for all clients, and the volume can therefore amend those listings to keep them consistent.

The CLI checks the volume's enabled shares for ABE consistency, and prompts for confirmation if any shares have ABE enabled at the back-end filer. The prompt warns you of the dangers of ABE inconsistency; a priv-exec command described in the next section can resolve this issue. For now, enter **yes** to proceed. (This confirmation prompt only appears if the volume has at least one enabled share; a later section describes the process of enabling the volume's shares.)

For example, the following command sequence disables ABE in the "medarcv~/lab_equipment" volume, which has enabled shares:

```
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# no cifs access-based-enum

This volume still contains at least one share with access-based enumeration
(ABE) enabled.  Disabling ABE on the volume without also disabling ABE on
all the volume's shares may result in user access problems.

Are you sure you want to disable ABE behavior for this volume? [yes/no] yes
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# ...
```

Disabling ABE at all Filer Shares Behind the Volume

After you disable ABE in the current volume, you should disable ABE on all of its back-end shares. For files on a back-end share that supports ABE, clients may see the following error in their directory listings:

```
02/24/2009  04:17 PM             9 myFile.doc
The parameter is incorrect.
```

This is due to the conflict between the ABE setting at the volume and at the back-end filer. From priv-exec mode, you can use the no form of the cifs access-based-enum command to disable ABE at every filer behind the volume:

◆ **Note**

As described above, this only operates on the currently-active node in a Windows Server Cluster. Use the cluster-administration interface or the abecmd.exe utility to consistently disable ABE on every node in the cluster. Alternatively, you can trigger a failover in the Windows cluster and re-run this command.

no cifs access-based-enum *namespace volume* [force]

where

namespace (1-30 characters) identifies this volume's namespace,

volume (1-1024 characters) is the path name for this volume, and

force (optional) causes the operation to work on all shares in the volume, including those that are not yet enabled.

The CLI prompts for confirmation before disabling ABE on any filers; enter **yes** to proceed. The CLI then produces a report to show the results of each ABE-disable operation: the report name and a summary of the results appears after you confirm. You can use show reports *report-name* to see detailed results.

For example, the following command sequence exits to priv-exec mode and disables ABE on all shares behind the "medarcv~/lab_equipment" volume:

```
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# end
bstnA# no cifs access-based-enum medarcv /lab_equipment

Volume medarcv~/lab_equipment is not configured for access-based enumeration (ABE).
Disabling ABE on this volume's shares will bring them into agreement
with the volume configuration.

Proceed to disable ABE for this volume's shares? [yes/no] yes

% INFO: Changed access-based enumeration settings for 3 of 3 shares. There were 0 errors and 0
shares were left unchanged. See report 'cifsAbeChange_200903031015.rpt'.

bstnA# ...
```

Adding ABE to a Running Volume

You can add ABE support to a running volume, after a back-end filer upgrade or a new decision to start using ABE. To accomplish this, you can enable ABE at the volume (as described previously) and then enable it at each of the volume’s back-end shares. You can enable ABE at the filers with Microsoft’s *abecmd.exe* utility, or by using this command from priv-exec mode:

```
cifs access-based-enum namespace volume [force]
```

where

namespace (1-30 characters) identifies this volume’s namespace,

volume (1-1024 characters) is the path name for this volume, and

force (optional) causes the operation to work on all shares in the volume.

This is the affirmative form of the no cifs access-based-enum command described previously. This enables ABE at all of the volume’s back-end shares, including any CIFS subshares. It generates a report to record its progress; use the show reports type AbCh command to list all ABE-change reports, or use show reports *report-name* to examine a report’s contents. The name of the report appears after you enter the command, along with a high-level summary of the results.

Note

This command can be used to force a back-end share’s ABE status to be consistent with the share’s ABE status in the ARX volume.

Note

As described previously, this only operates on the currently-active node in a Windows Server Cluster. Use the cluster-administration interface or the abecmd.exe utility to consistently enable ABE on every node in the cluster. Alternatively, you can trigger a failover in the Windows cluster and re-run this command.

After you enable ABE in the volume, the volume enforces ABE consistency on any new shares or subshares. The enable command (gbl-ns-vol-shr) will execute but will not actually succeed for a share if its ABE setting differs from that of the volume. The auto-enable flag addresses this by enabling ABE on the filer at share-import time.

For example, the following command sequence enables ABE in the “medarcv~/rcrds” volume, then exits to priv-exec mode and enables ABE on all of the volume’s shares and subshares:

```
bstnA(gbl-ns-vol[medarcv~/rcrds])# cifs access-based-enum auto-enable force

This volume still contains at least one share that does not have access-based
enumeration (ABE) enabled.  Enabling ABE on the volume without also enabling
ABE on all the volume's shares may cause confidential directory and file
names to be revealed to non-privileged personnel.

Are you sure you want to enable ABE behavior for this volume? [yes/no] yes
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# end
bstnA# cifs access-based-enum medarcv /rcrds
```

```
% INFO: Changed access-based enumeration settings for 6 of 6 shares. There were 0 errors and 0
shares were left unchanged. See report 'cifsAbeChange_200903031059.rpt'.

bstnA# ...
```

If a share has been enabled once already, use the priv-exec mode CLI command cifs access-based-enum <ns> <volpath> command to force consistency between the configuration and the back-end shares.

Adding a Share

The next step in creating a managed volume is identifying one or more shares for it. A *share* is one CIFS share or NFS export from an external (NAS or DAS) filer. A volume can contain multiple shares, where each typically resides on a different filer.

◆ **Note**

The first configured share in a managed volume, by default, will hold all new files created in the volume's root. The volume's root is the volume's top-most directory (for example, /acct). The share is called the root-backing share for the volume. Choose the root-backing share carefully, as it could possibly be overburdened by new files from clients.

To ease this burden, you can configure file-placement rules to distribute new files among multiple shares. You will learn how to configure these policies in a later chapter.

As with direct volumes, you use the share command to add a share to a managed volume. This puts you into gbl-ns-vol-shr mode, where you must identify the filer and export/share, and then you must enable the share. A managed-volume share also has several options related to import.

For example, this command set adds a share called "bills" to the "/acct" volume in the "wwmed" namespace:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# share bills
This will create a new share.

Create share 'bills'? [yes/no] yes
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Showing Available Filer Shares

You need to know the names and protocols of your configured filer shares to import one into your managed volume. Use the show exports external-filer ... shares command to show a filer's shares (recall *Listing Filer Shares*, on page 8-8), or use the show exports host ... shares command described in *Chapter 5, Examining Filers*. The first syntax identifies the filer by its external-filer name, the second uses the filer's IP address.

For example, the following command shows that there is only one NFS share on the “das8” external filer:

```
bstnA(gbl)# show exports external-filer das8 shares
Export probe of filer "das8" at 192.168.25.25
% INFO: Filer 192.168.25.25 does not support CIFS or is unreachable.

CIFS Credentials:
    User             (anonymous)

Shares:
  NFS
                                                 Read   Write     Space
    Path (Owner)                        Status   Size   Size   Total  Avail   FSID   Access
    --------------------------------    ------- ------ ------ ------ ------ -------- ------
    /work1                              Mounted  32 kB  32 kB  70 GB  27 GB fd00     *
bstnA(gbl)# ...
```

Identifying the Filer and Share

The most important step in configuring a share is connecting it to an export/share on an external filer. The export/share must support all of the namespace’s protocols; a CIFS namespace can only import CIFS shares, and an NFSv3 namespace can only import NFSv3 exports.

You use the filer command to identify the filer share behind the managed share. This is the same as the filer command for a direct-volume share (recall *Identifying the Filer and Share*, on page 8-10).

For example, this command set identifies an export for the “bills” share. The export is /work1/accting, on the “das8” filer:

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# filer das8 nfs /work1/accting
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Identifying a Multi-Protocol Share

For a multi-protocol namespace, you list both names for the share. You can do this in any order:

filer *name* nfs *nfs-name* cifs *cifs-name* [access-list *list-name*]

or

filer *name* cifs *cifs-name* nfs *nfs-name* [access-list *list-name*]

For example, the following command sequence uses a multi-protocol filer for the “insur~/claims~shr1-old” share:

```
bstnA(gbl)# namespace insur volume /claims share shr1-old
bstnA(gbl-ns-vol-shr[insur~/claims~shr1-old])# filer nas1 cifs insurance nfs
/vol/vol1/NTFS-QTREE/insurance
bstnA(gbl-ns-vol-shr[insur~/claims~shr1-old])# ...
```

Disconnecting From the Filer Before the Share is Enabled

To correct a mistake, you can disconnect a share from its filer before you enable the share. (The process of enabling a share is described later.) Use the no filer command described earlier in *Disconnecting From the Filer*, on page 8-12: For example, this command set disconnects the filer from the "spreadsheets" share:

```
bstnA(gbl)# namespace wwmed volume /acct share spreadsheets
bstnA(gbl-ns-vol-shr[wwmed~/acct~spreadsheets])# no filer
bstnA(gbl-ns-vol-shr[wwmed~/acct~spreadsheets])# ...
```

Disconnecting From the Filer After the Share is Enabled

After you enable the share, it has files and directories that were *imported* from the filer. That is, files and directories were scanned and incorporated into the volume's metadata. The *ARX CLI Maintenance Guide* describes how to remove an already-imported share without disrupting any client access to the share's files or directories.

Setting Share-Import Options

Each managed-volume share has several options that you can set to control its import. These subsections describe each of them in detail.

Speeding Up Import by Skipping a Directory Test

A managed volume tests each imported directory to verify that it is not already imported into another managed volume. Two volumes cannot manage the same directory. This managed-volume check is crucial for shares that may have been previously imported by any ARX, so it is enabled by default. For a new filer, the first introduction of the ARX at the site, or an nsck *rebuild* of the volume (described in the *ARX CLI Maintenance Guide*), you can disable this check to speed up the import of the share. From gbl-ns-vol-shr mode, use import skip-managed-check to skip the directory check:

```
import skip-managed-check
```

For example, the following command sequence allows the "medarcv~/lab_equipment" volume to skip this check while importing the 'equip' share.

```
bstnA(gbl)# namespace medarcv volume /lab_equipment share equip
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~equip])# import skip-managed-check
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~equip])# ...
```

Reinstating the Directory Test

If there is any doubt about any directory in the share, the volume should verify that none of them are managed by some other volume. Use the no import skip-managed-check command to re-instate the directory check:

```
no import skip-managed-check
```

For example, the following command sequence sets the /acct volume to check all the directories in the ‘bills’ share.

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# no import skip-managed-check
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Setting the Share’s Priority (for Tiering)

The volume compares its shares’ *import priorities* to determine which share wins a file or directory collision. A file collision occurs if a share contains a file with the same name and path as an already-imported file. Similarly, two directories collide if they have the same name and path but different attributes (such as read and write permissions). Whenever two shares have a collision, the file or directory that is imported first becomes the *master* instance of that object, and the second file or directory must change for the import to succeed. The master instance does not change. You can use the import priority command to ensure that your Tier 1 shares win any such collisions, and do not change on import:

import priority *number*

where *number* (1-65535) establishes the priority for this share. 1 is the highest priority and 65535 is the lowest. By default, all shares are at the lowest priority.

This priority persists during a re-import of the volume, if a re-import is ever necessary. This ensures that high-priority shares keep their masters if the volume ever requires a rebuild. For more information on volume rebuilds, see *Rebuilding a Volume*, on page 7-33 of the *ARX CLI Maintenance Guide*.

◆ **Important**

This is strongly recommended for a tiered volume. A tiered volume contains Tier-1 shares for the volume’s most-important files and Tier-2 shares for the remaining files. Best practices dictate that Tier 1 shares contain all of the volume’s master directories. We strongly recommend setting a priority of 1 (one) for all shares that you plan to use in Tier 1. (Tiering is discussed further in the policy chapters, later in this manual.)

For example, the following command sequence sets a priority of 1 (highest priority) for two shares in the ‘medarcv~/lab_equipment’ volume:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment share equip
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~equip])# import priority 1
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~equip])# exit
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# share leased
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~leased])# import priority 1
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~leased])# ...
```

If you do not set an import priority for any other share in the volume, this ensures that the “equip” and “leased” shares will win all import conflicts against other shares in the volume.

Reverting to the Default Import Priority

You can use no import priority to return the current share to its default priority:

```
no import priority
```

The default is the lowest possible priority, 65535, so this makes it unlikely that the share will win any collision conflicts after the next import. You should use this setting for Tier-2 or lower shares only, or for any share in a volume without any tiers.

For example:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment share backlots
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~backlots])# no import priority
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~backlots])# ...
```

Effect of Import Priority on Other Import Settings

The next sections describe commands to resolve import collisions, which may be irrelevant for some shares. If only a single share has the highest import priority in the volume, it wins all import conflicts and does not require any of these other import settings. The remaining shares, however, require these import commands to determine how their colliding files and directories should change.

All shares require the import settings if there are multiple shares with the highest priority. If these shares have colliding files and directories, the first-imported file or directory wins the conflict. This divides the master files and directories between your Tier-1 shares.

Synchronizing Directory Attributes on Import

Two directories collide when they have the same name but different file attributes, such as permission settings. If the managed volume has modify enabled (see *Allowing the Volume to Modify on Import*, on page 9-6), it renames any directory that collides with an already-imported directory. For an individual share, you can choose an alternative: instead of renaming the directory, synchronize its attributes with that of its already-imported counterpart. The volume presents the two directories as a single directory, with the aggregated contents of both and the attributes of the one that was imported first.

◆ **Note**

For heterogeneous multi-protocol namespaces, always enable synchronization with the import sync-attributes *command. A multi-protocol volume compares both CIFS and NFS attributes, thereby dramatically increasing the likelihood of directory collisions and renames.*

For a directory that collides with an already-imported *file*, a rename is the only possible method for resolving the conflict. This option cannot resolve a directory/file collision.

From gbl-ns-vol-shr mode, use the import sync-attributes command to allow directory-attribute synchronization for the current share.

```
import sync-attributes
```

For example, the following command sequence allows the /acct volume to synchronize directory attributes (instead of renaming conflicting directories) in the ‘bills’ share.

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# import sync-attributes
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Disabling Directory-Attribute Synchronization on Import

By default, if two shares have matching directories but conflicting attributes, the directories *collide*. The volume handles the collision by either renaming the second directory or failing the import, as specified by the modify flag. Some installations may prefer renamed directories to overwritten attributes. For situations where renamed directories are preferable, you can use the no import sync-attributes command to disable attribute synchronization:

```
no import sync-attributes
```

For example, the following command sequence turns off attribute synchronization in the ‘medarcv~/rcrds~bulk’ share during an import:

```
bstnA(gbl)# namespace medarcv volume /rcrds share bulk
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~bulk])# no import sync-attributes
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~bulk])# ...
```

◆ **Note**

A rename is impossible at the root directory of a share. If a new share’s root directory has attributes that are out of sync with the volume, the above setting would cause the import to fail.

Preventing Directory Renames During Import

The ability to change attributes does not solve the problem of directories that collide with previously-imported *files*. Attribute synchronization is not enough to resolve a naming conflict between a directory and a file. The renamed directory follows the same convention described earlier; recall *Allowing the Volume to Modify on Import*, on page 9-6.

Each share generates a report while it imports, and this report includes the original name and the new name for each renamed directory.

You can use the no import rename-directories command to protect this share against any directory renames.

```
no import rename-directories
```

If you set this, a directory/file collision causes the share to fail its import, and a directory/directory collision fails the import *unless* the volume is allowed to synchronize the attributes of the directories (as shown above). The *ARX CLI Maintenance Guide* explains how to remove a share from a namespace if its import fails; after that, you can directly connect to the back-end share and repair the directories that collided (change names, reset attributes, and so forth).

For example, the following command sequence prevents the volume from renaming any colliding directories in the ‘bills’ share during an import:

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# no import rename-directories
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Allowing Directory Renames on Import

If the share allows directory renames, the volume renames its colliding directories as specified by the modify command (refer back to *Allowing the Volume to Modify on Import*, on page 9-6). From gbl-ns-vol-shr mode, use the import rename-directories command to permit the volume to rename this share’s directories:

```
import rename-directories
```

This is the default setting.

For example, the following command sequence returns the ‘bills’ share to its default:

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# import rename-directories
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Renaming Directories with Non-Mappable Characters (Multi-Protocol)

This section applies to volumes in a multi-protocol (CIFS and NFS) namespace only. Skip to the next section if the volume supports only CIFS or only NFS.

If directory renames are allowed (as described above), you can set an additional option for renaming directories with non-mappable characters. (Non-mappable characters are discussed in the namespace chapter; see *Setting Up Character Encoding*, on page 7-11.) By default, any directory with a non-mappable character in its name causes the share import to fail. This default prevents accidental renames. You can ensure a successful import by allowing the volume to rename the directories. The volume uses same basic renaming syntax as with any other directory, but replaces each non-mappable CIFS character with its numeric Unicode equivalent:

new-dirname_share-jobid[-*index*][.*ext*]

> where *new-dirname* contains “(U+*nnnn*)” in place of each non-mappable character. The *nnnn* is the Unicode number for the character, shown in hexadecimal format. The name is truncated if it exceeds 256 characters. For example, “dir(U+30D2)(U+30AA)_myshare-2.”

The resulting name is visible through NFS and CIFS, and can be correlated to the intended CIFS name for the directory. As mentioned above, you can look at the share’s import report to see the original name and the new name for each renamed directory.

Use the unmapped-unicode option with the import rename-directories command to allow the volume to rename directories with non-mappable characters:

```
import rename-directories unmapped-unicode
```

For example, the following command sequence allows the "insur~/claims" volume to rename any directories with non-mappable characters:

```
bstnA(gbl)# namespace insur volume /claims share shr1-old
bstnA(gbl-ns-vol-shr[insur~/claims~shr1-old])# import rename-directories unmapped-unicode
bstnA(gbl-ns-vol-shr[insur~/claims~shr1-old])# ...
```

Preventing File Renames During Import

If the volume is allowed to modify files and directories on import, it renames files that collide with previously-imported files or directories. You can prevent the volume from renaming files in this share, causing the share's import to instead fail on file collisions. From gbl-ns-vol-shr mode, use the no import rename-files command to prevent any file renames in the current share:

```
no import rename-files
```

The *ARX CLI Maintenance Guide* explains how to remove a share from a namespace if its import fails (see *Removing an Imported Share*, on page 10-18 of that manual). Once the share is removed from the namespace, you can directly connect to the back-end share and repair the files that collided (change names, reset attributes, and so forth).

For example, the following command sequence disallows file renames in the 'bills' share during an import.

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# no import rename-files
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Allowing File Renames in Import

If the share allows file renames, the volume renames its colliding files as specified by the modify command.

```
import rename-files
```

This is the default setting.

For example, the following command sequence returns the 'bills' share to its default:

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# import rename-files
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Enabling SID Translation for a Share (CIFS)

This section only applies to a share that supports CIFS; skip to the next section if this share is in an NFS-only namespace.

A Windows filer can support *Global Groups*, which are managed by domain controllers, and/or *Local Groups*, which are unique to the filer. Local groups have their own Security IDs (SIDs), unknown to any other Windows machine. When you aggregate shares from these filers into a single volume, some files tagged for local-group X are likely to migrate to another filer, which does not recognize the SID for that group (SID X). This invalidates any Access Control Entries (ACEs) with SID X while the file resides on that filer; members of group X lose their privileges. To resolve this problem, you must first prepare all of the filers before you aggregate them into a namespace volume. This was discussed earlier; recall *Supporting Filers with Local Groups*, on page 9-19.

Once the filers are prepared, you can use the CLI to tag their shares for SID translation. This causes the volume to translate SIDs for all files that migrate to or from these shares: it finds the group name at the source share (such as “nurses” on filer B), then looks up the SID for that group name (“nurses”) at the destination share (on filer A).

Each share whose filer uses Local Groups must have SID translation enabled. For each of these shares, enter gbl-ns-vol-shr mode and use the sid-translation command:

```
sid-translation
```

For example, the following command sequence configures one share in the /rcrds volume for SID translation:

```
bstnA(gbl)# namespace medarcv volume /rcrds share bulk
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~bulk])# sid-translation
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~bulk])# ...
```

Disabling SID Translation

Use no sid-translation to stop the volume from translating SIDs for the share. This implies that the share is backed by a filer that uses global SIDs from the DC.

```
no sid-translation
```

For example, the following command sequence disables SID translation for the “rx” share:

```
bstnA(gbl)# namespace medarcv volume /rcrds share rx
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~rx])# no sid-translation
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~rx])# ...
```

Finding SID Translations at All Filers

From gbl-ns-vol mode, use the show sid-translation command to show the mapping between SIDs and names on all of the volume’s shares.

```
show sid-translation principal
```

where ***principal*** (1-256 characters) is a group name, user name, or SID to translate.

The output displays the translation at each share. For example, the following command sequence discovers that the shares behind the ‘medarcv~/rcrds’ volume have different SIDs for the ‘pharmacists’ group:

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# show sid-translation pharmacists
SID Translations:

  Share rx
    SID:  S-1-5-21-1454471165-492894223-682003330-1006
    Name: MEDARCH\pharmacists (group)
  Share charts
    SID:  S-1-5-21-1454471165-492894223-682003330-1006
    Name: MEDARCH\pharmacists (group)

  Share bulk
    SID:  S-1-5-21-2006398761-1897008245-3502739112-1007
    Name: PV770N\pharmacists (alias)

bstnA(gbl-ns-vol[medarcv~/rcrds])# ...
```

Ignoring SID Errors from the Filer (CIFS)

If the share does not perform SID translation, or if SID translation fails, the share copies the binary version of the SID to the destination filer. It is possible that the SID is unknown at the destination filer. Typically, the back-end filer returns an error and rejects the file, or silently accepts the unknown SID: an error indicates that the file was rejected. The share therefore aborts the migration if it receives any of the following errors in response to the CIFS ‘set security descriptor’ command: STATUS_INVALID_SID, STATUS_INVALID_OWNER, or STATUS_INVALID_PRIMARY_GROUP.

Some file servers issue these errors for unknown SIDs but accept the file anyway. Some EMC file servers have this setting as a default. As long as the file server is configured to accept the file or directory (perhaps erasing the unknown SIDs), the volume can safely ignore these errors.

◆ **Important**

Do not ignore any SID errors from a file server that rejects the file or directory. The SID errors alert the volume to the failed migration. If the volume ignores SID errors from such a file server, file and/or directory loss may result.

To ignore SID errors from the file server, use the ignore-sid-errors command from gbl-ns-vol-shr mode:

```
ignore-sid-errors
```

For example, the following command sequence ignores all SID errors from the file server behind the “insur~/claims~shr1-next” share:

```
bstnA(gbl)# namespace insur volume /claims share shr1-next
bstnA(gbl-ns-vol-shr[insur~/claims~shr1-next])# ignore-sid-errors
bstnA(gbl-ns-vol-shr[insur~/claims~shr1-next])# ...
```

Acknowledging SID Errors

It is unsafe to ignore SID errors from a filer or file server that is properly configured. If SID errors truly indicate that the file was rejected, the ARX share should acknowledge them and cancel the file migration. To acknowledge SID errors from the filer, use the no ignore-sid-errors command:

```
no ignore-sid-errors
```

For example, the following command sequence acknowledges all SID errors from the filer behind the “insur~/claims~shr1-old” share:

```
bstnA(gbl)# namespace insur volume /claims share shr1-old
bstnA(gbl-ns-vol-shr[insur~/claims~shr1-old])# no ignore-sid-errors
bstnA(gbl-ns-vol-shr[insur~/claims~shr1-old])# ...
```

Excluding the Share from ABE Support (CIFS)

This section only applies to a volume where ABE is enabled, as described earlier; recall *Supporting Access-Based-Enumeration (ABE)*, on page 9-27. Skip to the next section unless this volume supports ABE and contains at least one share that you want to exclude from all ABE processing.

You have the option to exclude this share from ABE processing, though it is not generally recommended. This option is designed for volumes with tiered storage, where a second or third tier cannot support ABE due to the filer(s) used. For example, this could apply to a site backed by Samba filers for tier-3 storage, where files migrate to tier 3 if they remain unchanged for months. The disadvantage of using this option is that a file suddenly appears to some clients after it migrates to the ABE-disabled share, and the clients that now see it are the same clients who have no permission to read it.

◆ **WARNING**

Avoid all policies that mix ABE-enabled shares with ABE-disabled shares. Put all ABE-disabled shares in the volume’s lowest tier, and do not mix any ABE-enabled shares in the same tier. The rules that enforce tiering (described in a later chapter) should be the only rules in the volume that migrate to or from the ABE-disabled shares.

From gbl-ns-vol-shr mode, you can use the cifs access-based-enum exclude command to inform the volume that this share does not support ABE. This makes it possible to enable the share in a volume that otherwise supports ABE:

```
cifs access-based-enum exclude
```

For example, the following command sequence excludes the “scanners” share from ABE support in the “medarcv~/lab_equipment” volume:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment share scanners
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~scanners])# cifs access-based-enum exclude
bstnA(gbl-ns-vol-shr[medarcv~/lab_equipment~scanners])# ...
```

Supporting ABE at the Share

After a filer change, you may be able to support ABE on a share that formerly could not. You can remove the ABE-support exclusion with the no form of the above command:

```
no cifs access-based-enum exclude
```

Remember to enable ABE at the back-end share before you remove this exclusion.

For example, the following command sequence adds ABE support to the formerly-excluded "medarcv~/rcrds~t3_2" share:

```
bstnA(gbl)# namespace medarcv volume /rcrds share t3_2
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~t3_2])# no cifs access-based-enum exclude
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~t3_2])# ...
```

Designating the Share as Critical (optional)

If the current switch has a redundant peer, you have the option to designate the current share as *critical*. Skip to the next section if this switch is not configured for redundancy.

If the volume software loses contact with one of its critical (and enabled) shares, the ARX initiates a failover. The failover is accepted by the redundant peer as long as the peer has full access to all critical shares, critical routes, *and* the quorum disk. If the peer is unable to access any of these critical resources, no failover occurs.

As explained in the direct-volume chapter (recall *Designating the Share as Critical (Optional)*, on page 8-14), you can use the critical command to designate the current share as a critical resource. For example, this command sequence designates the "bills" share as a critical share:

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# critical
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Removing Critical-Share Status

By default, shares are not critical. If the switch loses contact with a non-critical share, the volume operates in a degraded state and the switch does not initiate a failover. Use the no critical command described earlier (*Removing Critical-Share Status*, on page 8-15). For example, this command sequence removes the "bills2" share from the list of critical shares:

```
bstnA(gbl)# namespace wwmed volume /acct share bills2
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills2])# no critical
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills2])# ...
```

Enabling the Share

The final step in configuring a share is to enable it. An enabled share is an active part of the managed volume; the volume uses enabled shares in its storage pool, and excludes all disabled shares. As with a direct-volume share, use the enable command in gbl-ns-vol-shr mode.

◆ **Note**

If this is a CIFS volume and it replicates subshares (recall Replicating Subshares at all of the Volume's Filers, on page 9-23), all subshares are copied to any shares that you enable later. If you want any subshares to be used in the volume, access the back-end filer directly and add them before you enable the first share.

For example, the following command sequence enables the "wwmed ~/acct~bills" share.

```
bstnA(gbl)# namespace wwmed volume /acct share bills
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# enable
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# ...
```

Enable all of the volume's shares before you enable the managed volume. The best practice is to allow the managed volume to import all shares simultaneously.

Taking Ownership of the Share (optional)

Before a managed volume imports the share, it checks the root directory in the back-end share for a special file that marks it as "owned" by an ARX. If this marker file exists, the managed volume does not proceed with the import; no two volumes can manage the same share. You may need to override this safety mechanism for a special case.

Consider an installation that uses legacy, filer-based applications to prepare for disaster recovery: it copies all of its directories and files from a primary site to the filers at another site. If an ARX manages the directories at the primary site, it places its ownership marker in the root of each share. The filer-based application copies the marker files to the remote site, along with all data files. An ARX at the backup site cannot import these shares because of the markers.

You can use the optional take-ownership flag for this special case. If the managed volume finds an ownership marker in the root of this back-end share, it overwrites the marker file. Otherwise, it imports the share as usual:

```
enable take-ownership
```

◆ **Important**

Do not use this option if it is possible that another ARX is managing this back-end share. This would unexpectedly remove the share from service at the other ARX.

The CLI prompts for confirmation before taking ownership of the share. Enter **yes** to proceed.

For example, the following command sequence enables the “insur_bkup~/insurShdw~backInsur” share, taking ownership of the share if necessary:

```
prtlndA(gbl)# namespace insur_bkup volume /insurShdw share backInsur
prtlndA(gbl-ns-vol-shr[insur_bkup~/insurShdw~backInsur])# enable take-ownership
This command allows the switch to virtualize shares that are used by other Acopia switches.
Allow switch to take ownership of share? [yes/no] yes
prtlndA(gbl-ns-vol-shr[insur_bkup~/insurShdw~backInsur])# ...
```

Examining the shareEnableSubshareInc Report (CIFS)

This only applies to CIFS volumes where filer-subshares is enabled; recall *Supporting Subshares and their ACLs*, on page 9-20. Skip this section unless this option is active in the volume.

Whenever you enable a share in such a CIFS volume, the volume generates a report on all subshare-related activity. The CLI shows the name of the report after you issue the enable command. For example:

```
bstnA(gbl)# namespace medarcv volume /rcrds share charts
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~charts])# enable

% INFO: Successfully replicated all consistent back-end subshares to 'charts'. See report
'shareEnableSubshareInc_200709240243.rpt' for details.

bstnA(gbl-ns-vol-shr[medarcv~/rcrds~charts])# ...
```

This report shows the progress of any replications or subshare-consistency checks that the volume performed. Use the show reports *report-name* command to view the report. For example, this shows the report from the above share-enable command:

```
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~charts])# show reports shareEnableSubshareInc_200709240243.rpt
**** Share Enable Filer-Subshare Inconsistency Report: Started at Mon Sep 24 02:43:46 2007 ****
**** Software Version: 3.00.000.10541 (Sep 21 2007 18:11:56) [nbuilds]
**** Hardware Platform: ARX-6000

The following changes were made to replicate nested shares and their
attributes to the new share:

  Added share “_acopia_CELEBS$_1$” to the following filer:

    Filer Name: fs1
    IP Address: 192.168.25.20
    Path:       d:\exports\histories\VIP_wing


  Added share “_acopia_Y2004_2$” to the following filer:

    Filer Name: fs1
    IP Address: 192.168.25.20
    Path:       d:\exports\histories\2004


  Added share “_acopia_Y2005_3$” to the following filer:

    Filer Name: fs1
    IP Address: 192.168.25.20
    Path:       d:\exports\histories\2005

**** Total processed:              3
```

```
**** Elapsed time:          00:00:04
**** Share Enable Filer-Subshare Inconsistency Report: DONE at Mon Sep 24 02:43:50 2007 ****
bstnA(gbl-ns-vol-shr[medarcv~/rcrds~charts])# ...
```

This sample report shows that the volume successfully replicated three subshares. Later, all three subshares can be exported from a front-end CIFS service.

Disabling the Share

You can disable a share to make it inaccessible to namespace clients. This stops access to all files on the share, as well as all directories with their masters on the share. (If a directory contains files on multiple shares, due to file migrations, the first-imported share is said to have the *master* copy of the directory. This is called the *master directory*.)

This is not generally recommended in a managed volume. Use this to prepare for back-end filer maintenance (just before taking the filer offline), and only on a share with very few master directories.

As in a direct volume, use no enable in gbl-ns-vol-shr mode to disable the share.

◆ **Important**

This suspends all policy rules in the current volume; the rules are enabled, but not enforced. To bring policy back online for the current volume, remove the share (described below) or re-enable it.

Removing a Managed-Volume Share

You can easily remove a share before its volume is first enabled and starts importing files. Use no share in this case, just as describe for a share in a direct volume. For example, this command set removes the “billsBU” share from the “/acct” volume in the “wwmed” namespace:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# no share billsBU
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Removing the Share After it is Enabled

After you enable the share, it has files and directories that were imported from the filer. That is, files and directories were scanned and incorporated into the volume’s metadata. These files are visible to the volume’s clients, and cannot be removed without disrupting client service. A client-friendly approach is to migrate the files to other shares in the same volume; clients can access the files throughout the migration. The *ARX CLI Maintenance Guide* explains how to migrate all the files and remove an imported share with a single command (see *Removing an Imported Share*, on page 10-18 of that manual).

Assigning The Volume To A Volume Group (optional)

The next step in configuring a volume is to assign it to a volume group. The ARX's namespace functionality will assign the volume to a default volume group automatically if you do not specify one explicitly.

◆ **Note**

Volume group assignments become permanent once a volume is enabled.

Volume groups are a means of isolating namespaces in the ARX's memory so that the failure of one or more namespaces in one volume group does not affect the performance of namespaces in other volume groups. Volume groups are a virtual resource that can fail over from one ARX to another in a redundant configuration.

The maximum number of volume groups supported per ARX model is a function of the ARX's memory, and can be between 2 and 24, depending on the ARX model. Within that range of possible volume groups is the number of volume groups that are allowed by the particular license agreement that the customer has in place for a specific ARX instance. This number of volume groups may be less than the maximum, depending on the number that the customer has paid to license, but will never exceed the maximum number supported for that model. Follow this link to a table that shows the maximum number of volume groups supported by each ARX model:

http://support.f5.com/content/dam/f5/kb/global/solutions/sol13129_images.html/Platform_matrix_ARX_system_limits_enforced_by_licensing.pdf

◆ **Note**

For the ARX-500, two volume groups are supported by default, and a CLI command can be used to increase the number of volume groups to the maximum number supported for that model, which is four. This is an artifact of the hardware design of this model.

Assigning a Volume To a Volume Group Explicitly

When you are configuring a volume and are in gbl-ns-vol mode, you can assign the volume to a specific volume group by using the volume group CLI command. The syntax is:

```
volume-group volumegroupid
```

where *volumegroupid* is the integer that identifies the volume group to which you are assigning the current volume.

For example:

```
bstnA(gbl-ns-vol[bgh~/naumkeag_wing])# volume-group 2
```

assigns the current volume, “naumkeag_wing”, to volume group 2.

Migrating a Volume to a Different Volume Group

As your deployment grows, you may want to redistribute volumes among volume groups to mitigate possible down time. You can move one or more volumes from one volume group to another by using the nsck command with the migrate-volume option. This is done in privileged-exec mode.

The command syntax is:

```
nsck name migrate-volume volname volume-group id [check-limits]
```

where:

name is the namespace within which or to which the volume will be moved;

volname specifies the volume to be moved; and

id specifies the volume-group to which the volume will be moved.

The check-limits argument is optional, and enables you to verify that the volume migration will work without actually initiating it. It checks the licensed limits at the destination volume group, and for any other possible obstacles that might prevent a successful migration.

The example that follows specifies a volume to migrate, but uses the check-limits option to verify it first:

```
bstnA# nsck medarcv migrate-volume /test_results volume-group 3 
check-limits
```

Then, execute the command without the check-limits argument to initiate the migration of the volume to a different volume group:

```
bstnA# nsck medarcv migrate-volume /test_results volume-group 3
Volume /test_results is in use by CIFS global service 
'ac1.medarch.org'.
Volume /test_results in namespace medarcv is in use or being browsed 
by global services.
Migrating the volume to a different Volume-Group will disrupt all 
clients accessing this volume through these services.
No other volume will be affected by this procedure.
Proceed with volume migration? [yes/no] yes
Scheduling report: migrate_volume_group.35.rpt on switch bstnA
```

To cancel the volume migration (and return the volume to service in its original volume group), use cancel migrate-volume.

The nsck migrate-volume command stops client access to that volume until the migration is complete. When the command is executed, the CLI lists all of the affected front-end services and prompts you for confirmation; type “yes” to continue with the migration.

A volume cannot be migrated while it is importing files from its back-end shares. The import must be complete before you can move the volume to another volume group.

Very large volumes may take a long time to finish migrating. Upon execution, the nsck migrate-volume command returns immediately, providing you with the name of a detailed report that will record the migration. The migration report is named “migrate_volume_group.*nsck-job-id*.rpt.” Use show reports to list all reports, including this one. Use the tail reports *report-name* follow command to follow the progress of the migration. You can continue using the CLI while the migration occurs in the background. The show nsck command shows the current status of the migration, along with the status of all other nsck operations. After the nsck job is complete, front-end services will require a few additional seconds before they allow client access to the volume.

Reverting to The Default Volume Group Assignment

Prior to enabling a volume, you can revert it to its default volume group assignment by using the no volume-group CLI command in gbl-ns-vol mode:

```
bstnA(gbl-ns-vol[bgh~/naumkeag_wing])# no volume-group
```

This undoes any explicit volume group assignment that you may have configured, and assigns the volume to whatever default volume group it would have been assigned automatically.

Showing Volume Groups On The Current ARX

Use the show volume-group CLI command to display all of the volume groups on the current ARX. For example:

```
bstnA# show volume-group


Switch: bstnA
----------------------------------------------------------------------
System Credits
--------------
Share credits: 4 shares used (380 credits remain of total 384)
Direct share credits: 1 direct shares used (6143 credits remain of
total 6144)
Volume credits: 2 volumes used (94 credits remain of total 96)
File credits: 4.0 M files reserved (764 M credits remain of total 768
M)


Volume Group 1
------------
Physical Processor: 1.1
State: Normal; partially used
Share credits: 4 shares used (188 credits remain of total 192)
Direct share credits: 1 direct shares used (3071 credits remain of
total 3072)
Volume credits: 2 volumes used (46 credits remain of total 48)
File credits: 4.0 M files reserved (380 M credits remain of total 384
M)
```

```
Namespace         Volume Group  Volume              State
---------         ------------  ------              -----
bgh                1      /naumkeag_wing       Enabled

1 Namespace             1 Volume
```

Display a more detailed set of information for the volume groups by appending the detailed argument to the command:

```
bstnA# show volume-group detailed

Switch: bstnA
----------------------------------------------------------------------
System Credits
--------------
Share credits: 4 shares used (380 credits remain of total 384)
Direct share credits: 1 direct shares used (6143 credits remain of
total 6144)
Volume credits: 2 volumes used (94 credits remain of total 96)
File credits: 4.0 M files reserved (764 M credits remain of total 768
M)

Volume Group 1
------------
Physical Processor: 1.1 (29% CPU, 40% MEM)
State: Normal; partially used
Share credits: 4 shares used (188 credits remain of total 192)
Direct share credits: 1 direct shares used (3071 credits remain of
total 3072)
Volume credits: 2 volumes used (46 credits remain of total 48)
File credits: 4.0 M files reserved (380 M credits remain of total 384
M)

Namespace         Volume Group  Volume              State
---------         ------------  ------              -----
bgh                1      /naumkeag_wing       Enabled

1 Namespace             1 Volume
```

Display information for a specific volume group by specifying a volume group ID as an argument to the command:

```
bstnA# show volume-group 1

Switch: bstnA
----------------------------------------------------------------------
System Credits
```

```
--------------
Share credits: 4 shares used (380 credits remain of total 384)
Direct share credits: 1 direct shares used (6143 credits remain of
total 6144)
Volume credits: 2 volumes used (94 credits remain of total 96)
File credits: 4.0 M files reserved (764 M credits remain of total 768
M)

Volume Group 1
------------
Physical Processor: 1.1
State: Normal; partially used
Share credits: 4 shares used (188 credits remain of total 192)
Direct share credits: 1 direct shares used (3071 credits remain of
total 3072)
Volume credits: 2 volumes used (46 credits remain of total 48)
File credits: 4.0 M files reserved (380 M credits remain of total 384
M)

Namespace         Volume Group  Volume              State
---------         ------------  ------              -----
bgh                1      /naumkeag_wing       Enabled

1 Namespace            1 Volume
```

Display detailed information for a specific volume group by specifying a volume group ID as an argument to the command:

```
bstnA# show volume-group 1 detailed

Switch: bstnA
----------------------------------------------------------------------
System Credits
--------------
Share credits: 4 shares used (380 credits remain of total 384)
Direct share credits: 1 direct shares used (6143 credits remain of
total 6144)
Volume credits: 2 volumes used (94 credits remain of total 96)
File credits: 4.0 M files reserved (764 M credits remain of total 768
M)

Volume Group 1
------------
Physical Processor: 1.1 (29% CPU, 40% MEM)
State: Normal; partially used
Share credits: 4 shares used (188 credits remain of total 192)
Direct share credits: 1 direct shares used (3071 credits remain of
total 3072)
```

```
Volume credits: 2 volumes used (46 credits remain of total 48)
File credits: 4.0 M files reserved (380 M credits remain of total 384
M)

Namespace         Volume Group  Volume               State
---------         ------------  ------               -----
bgh                1      /naumkeag_wing       Enabled


1 Namespace             1 Volume
```

where *volumegroupid* is the integer that identifies the volume group for which you want to display information.

Increasing the Number of Volume Groups Available (ARX-500)

On the ARX-500, the number of volume groups available for use can be increased from the default of two to the maximum supported for that ARX model, using the max-volume-groups CLI command in gbl mode.

The syntax is:

```
bstnA(gbl)# max-volume-groups
```

Executing this command increases the maximum number of volume to groups to the maximum number allowed for the current ARX model.

This command can be negated using the no max-volume-groups command, which returns the number of volume groups to the default of two.

Enabling the Volume

The final step in configuring a managed volume is to enable it. This is the same as for a direct volume: from gbl-ns-vol mode, use the enable command to enable the current volume.

The enable command starts the import of external files into all enabled shares. An import does not occur in a direct volume. This is a process of walking the directory tree on each back-end share, recording file locations, and checking against already-imported files for naming collisions. All of the managed volume's shares should be enabled before you enable the volume, so that the volume can import all of the shares simultaneously.

◆ **Important**

Clients cannot directly access the filers behind this volume during or after import. Block all client access to all back-end shares before you enable the managed volume.

◆ **Note**

Once the namespace's first import begins, the namespace protocol *becomes more difficult to change. Before you enable the first managed volume, recall* ***Changing Protocols After Import****, on page 7-10.*

For example, the following command sequence enables the "/acct" volume in the "wwmed" namespace:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# enable
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

The import happens asynchronously, so that you can issue more CLI commands and clients can access the volume's storage while the import happens in the background. To check the progress of the import, use show namespace [status], as described below in "*Monitoring the Import*."

Enabling All Shares in the Volume

From gbl-ns-vol mode, you can enable all of the volume's shares with a single command. As with a direct volume, you use the enable shares command to do this (refer back to *Enabling All Shares in the Volume*, on page 8-24). For example, the following command sequence enables all shares in the "/acct" volume:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# enable shares
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Taking Ownership of All Shares (optional)

Before a managed volume imports its shares, it checks the root directory in each share for a special file that marks it as “owned” by an ARX. If this marker file exists, the managed volume does not proceed with the import; no two volumes can manage the same share. You may need to override this safety mechanism for a special case.

Consider an installation that uses legacy, filer-based applications to prepare for disaster recovery: it copies all of its directories and files from a primary site to filers at another site. If an ARX manages the directories at the primary site, it places its ownership marker in the root of each share. The filer-based application copies the marker files to the remote site, along with all data files. An ARX at the backup site cannot import these shares because of the markers.

You can use the optional take-ownership flag for this special case. If the managed volume finds an ownership marker in the root of any of its shares, it overwrites the marker file. Otherwise, it imports the share as usual:

```
enable shares take-ownership
```

◆ **Important**

Do not use this option if it is possible that another ARX is managing one of the volume’s shares. This would unexpectedly remove the share(s) from service at the other ARX.

The CLI prompts for confirmation before taking ownership of any shares. Enter **yes** to proceed.

For example, the following command sequence enables all shares in the “insur_bkup~/insurShdw” volume and takes ownership of all of them:

```
prtlndA(gbl)# namespace insur_bkup volume /insurShdw
prtlndA(gbl-ns-vol[insur_bkup~/insurShdw])# enable shares take-ownership
This command allows the switch to virtualize shares that are used by other Acopia switches.
Allow switch to take ownership of all the shares in this volume? [yes/no] yes
prtlndA(gbl-ns-vol[insur_bkup~/insurShdw])# ...
```

Disabling All Shares

Use no enable shares command to disable each of the volume’s individual shares.

◆ **Important**

This is equivalent to disabling the volume, described below. This causes the volume to stop responding to clients; different client applications react to this in different ways. Some may hang, others may log errors that are invisible to the end user.

For example, this command sequence disables all of the shares in the “ns1~/vol” volume:

```
bstnA(gbl)# namespace ns1 volume /vol
bstnA(gbl-ns-vol[ns1~/vol])# no enable shares
bstnA(gbl-ns-vol[ns1~/vol])# ...
```

Disabling the Volume

You can disable a volume to stop clients from accessing it. Just as described with a direct volume, you disable the volume with no enable in gbl-ns-vol mode. (See *Disabling the Volume*, on page 8-24.)

For example, the following command sequence disables the "/acct" volume:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# no enable
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Monitoring the Import

A managed volume, unlike a direct volume, imports all back-end files and directories after you enable it. Use the show namespace status command to monitor the progress of an import:

```
show namespace status {namespace | all}

show namespace status namespace volume vol-path

show namespace status namespace volume vol-path share share-name
```

where:

namespace (1-30 characters) is the name of a namespace.

all displays details for all namespaces.

vol-path (1-1024 characters) narrows the output to a single volume.

share-name (1-64 characters) narrows the output further, to one share.

For example, the following command shows the status of the 'wwmed' namespace.

```
bstnA(gbl)# show namespace status wwmed

Namespace: wwmed
Description: namespace for World-Wide Medical network

     Share                     Filer                                 Status
         NFS Export
    ------------------------- ------------------------------------- -----------

  Volume: /acct                                                    Enabled
     budget                    das1                                  Online
         NFS: /exports/budget

     bills                     das8                                  Online
         NFS: /work1/accting

     bills2                    das3                                  Online
         NFS: /exports/acct2

     it5                       das7                                  Online
         NFS: /lhome/it5

     metadata-share            nas1                                  Online
```

```
        NFS: /vol/vol2/meta1

bstnA(gbl)# ...
```

The Status for each imported share should go through the following states:

1. Pending,
2. Importing, then
3. Online.

The show namespace command (see *Showing Namespace Details*, on page 7-4) shows more detail after the “Importing” flag:

Importing: *a* items scanned, *b* items imported

Import Errors

If anything goes wrong, the failed share shows a Status of “Error” in the show namespace status output, and a specific error appears in the more-verbose show namespace output. Use show namespace (not show namespace status) to see the full error message. To correct the error, refer to the *ARX CLI Maintenance Guide*.

Canceling a Share Import

From priv-exec mode, you can cancel the import of a single share with the cancel import command:

cancel import namespace *ns* volume *vol-path* share *share-name*

where:

ns (1-30 characters) identifies the namespace.

vol-path (1-1024 characters) is the share’s volume.

share-name (1-64 characters) is the share.

For example, the following command sequence stops the import of the “wwmed~/acct~expir” share:

```
bstnA(gbl)# end
bstnA# cancel import namespace wwmed volume /acct share expir
Storage job cancelled successfully.
bstnA# ...
```

Reviewing the Import Report for a Multi-Protocol Volume

A multi-protocol (NFS and CIFS) volume may import successfully but still have several issues of interest. These issues center on files and/or directories with filer-generated names (FGNs, such as “dir~2”) or other CIFS/NFS naming issues that cause the volume to declare them “NFS-only.” An *NFS-only* file or directory, as the name implies, cannot be removed, renamed, or edited by CIFS clients. CIFS clients can only view NFS-only entries in their directory listings. NFS clients and the policy engine have full access, though rare client operations and policy migrations will incur a performance penalty. For full details, see *Finding NFS-Only Entries in a Multi-Protocol Volume*, on page 10-64 of the *ARX CLI Maintenance Guide*.

Examine the import report for each share in the multi-protocol volume. Each share has a separate import report named "import.*report-id*.*share-name*.*share-id*.rpt." Use show reports type Imp to get a list of import reports, and show reports *report-name* to view the named report. Look for files or directories flagged with any Multi-Protocol-error flags. An NFS client can typically resolve these problems by renaming the file or directory through the managed volume.

For example, this shows the import report for the "shr1-old" share. The import contains several files and directories with multi-protocol issues, highlighted in bold text:

```
bstnA# show reports import.12.shr1-old.30.rpt
**** Share Import Report: Started at 04/11/2012 01:07:49 -0400 ****
**** Software Version: 6.02.000.14353 (Apr  6 2012 20:12:43) [nbuilds]
**** Hardware Platform: ARX-4000
**** Report Destination:

**** Namespace: insur
**** Volume:    /claims
**** Share:     shr1-old
**** IP Addr:   192.168.25.21
**** Export:    insurance
**** Export:    /vol/vol2/insurance
**** Options:
****            modify:                          yes
****            rename-directories:              yes
****            rename-non-mappable-directories: yes
****            rename-files:                    yes
****            sync-attributes:                 yes
****            strict-attribute-consistency:    no
****            skip-managed-check:              no
****            import protection:               yes
****            import treewalk-threads:         8
****            import priority:                 65535 (Lowest)

**** NOTE: Since both sync-attributes and rename-directories were specified,
****       only directories that collide with existing filenames will be
****       renamed.  Directories with colliding attributes will have their
****       attributes synchronized to the namespace.

**** Ignore List:
****            .clusterConfig
****            SIS Common Store
****            System Volume Information
****            $RECYCLE.BIN
****            .ckpt*
****            .etc
****            ~snapshot
****            .snapshot


Share                 Physical Filer
--------------------  ----------------------------------------------------------
[shr1-old          ]  192.168.25.21 NFS:/vol/vol2/insurance, CIFS:insurance
```

```
**** LEGEND
****
****  Actions
****   R   : Entry renamed.
****   S   : Directory attributes synchronized.
****   ?   : Indicates additional settings or operations required for full
****         operation.  See specific issues for SD and IN below.
****   !   : Unable to resolve given specified options or configuration.
****
****  Entry Type
****   F   : Entry is a file.
****   D   : Entry is a directory.
****
****  Issue
****   NC  : Name collision.
****   AC  : Attribute collision.
****   RA  : Attributes of share root are inconsistent.
****   MG  : Subdirectory of this share is already imported as managed share.
****   CC  : Case-blind collision (MPNS and CIFS-only).
****   MC  : Maximum name collisions for this name in a directory
****   ER  : Entry removed directly from filer during import.
****   AE  : Error accessing entry.
****   RV  : Reserved name on filer not imported.
****   DF  : DFS link found during import.
****   TC  : Trailing character: space and period are not supported by all filer vendors.
****   HL  : Exceeded limit of 1024 hard links to a file.
****   PF  : Directory promote failed and master directory is mis-located
****   DA  : Directory attributes sync failed and are inconsistent between shares
****   NF  : Specified Path was not found.
****   IG  : Directory ignored during import.
****   @@  : Other error.
****
****  Multi-Protocol Issue
****   NE  : Entry found with NFS that was not found with CIFS (is NFS-only).
****   CE  : Entry found with CIFS that was not found with NFS.
****   IC  : CIFS invalid characters found in NFS name.
****   NM  : CIFS name has characters that are not mappable to the NFS encoding.
****   U8  : invalid UTF8 characters found in name and entry not imported.
****   FC  : A filer-generated-name (FGN) collision is not supported.
****   IN  : Potential inconsistency was marked nfs-only.  If no-modify import,
****         run 'sync files' to resolve, or re-run import as 'modify'.
****   SD  : Unable to synchronize directory attributes due to a name inconsistency.
****         Directory will require "no strict-attr" to support migration.


Import Scan:
================================================================================

----------------------------------------------------------
Import Scan Start Time:   04/11/2012 01:07:59 -0400
----------------------------------------------------------

Type      Path
--------  ----------------------------------------------------------------------
[  F NM]  /images/file012bÄ«/ (Characters: U+012b)
[R D NM]  /dir0134Ä´ -> dir0134(U+0134)_shr1-old-12
[  F IC]  /images/FRU replace wrongway.tif
[  F IC]  /images/:2KQC000
[  F IC]  /stats/on_the_job:2004.cnv
[  F IC]  /stats/on_the_job:2003.cnv
[  F IC]  /stats/in_home:2005/age:11-21yrs.csv
[  F IC]  /stats/in_home:2005/age:>21yrs.csv
[  F IC]  /stats/in_home:2005/age:<10yrs.csv
[  F IC]  /stats/in_home:2005/age:>21yrs.csv
[  F IC]  /stats/in_home:2005/age:<10yrs.csv
[  F IC]  /stats/in_home:2005/age:11-21yrs.csv
[  D IC]  /Claims:2001
[  D IC]  /claims:2005
[  D IC]  /:7B6J000
[  D IC]  /draft_proposals.
[  D IC]  /stats/in_home:2005
[? F IN]  /INDEX.html
[? F IN]  /stats/piechart.ppt
[? D IN]  /Tools
[  F CC]  /INDEX.html
[  F CC]  /index.html
[  F CC]  /stats/piechart.ppt
[  F CC]  /stats/PieChart.ppt
[  D CC]  /Tools
[  D CC]  /tools

Directories found:                       19
Files found:                            142
Directories Scanned/Second:              19
Files Scanned/Second:                   142
Total Entries Scanned/Second:           161

----------------------------------------------------------
Import Scan Stop Time:    04/11/2012 01:07:59 -0400
Import Scan Elapsed Time: 00:00:01
----------------------------------------------------------

Directories imported by scan:                                     19
Directories imported dynamically:                                  0
Files imported by scan:                                          142
Files imported dynamically:                                        0
Directories renamed due to name/attribute conflict:                1
Files renamed due to name conflict:                                0
Directory attributes synchronized:                                 0
Files/directories with case-blind collisions:                      6
Files/directories with invalid CIFS characters:                   12
NFS files/directories with filer generated names:                  2
Files/directories found via only one protocol:                    11
CIFS files with non-mappable Unicode characters:                   1

**** Elapsed time:           00:00:10
**** Share Import Report: DONE at 04/11/2012 01:07:59 -0400 ****

bstnA# ...
```

Showing the Volume

The direct-volume chapter discussed some show commands that focus on volumes; recall *Showing the Volume*, on page 8-26. These same commands work on all volume types, including managed volumes. The difference is the output; managed volumes support policy (described in the next chapter), so any rules in the volume appear here.

To show only one volume in a namespace, add the volume clause to show namespace command. For example, the following command shows the configuration of the ‘medarcv~/rcrds’ volume:

```
bstnA# show namespace medarcv volume /rcrds

Namespace "medarcv" Configuration
Description: (none)
Metadata Cache Size: 512 MB
Proxy User: acoProxy2
Filer SMB Signatures: Enabled
SAM Reference Filer: fs2 (192.168.25.27)

Supported Protocols
-------------------
  cifs

CIFS Authentication
-------------------
Protocols:
  NTLM
  NTLMv2
  Kerberos

Participating Switches
----------------------
  bstnA (Volume Group 2) [Current Switch]


Windows Management Authorization Policies
----------------------------------------
  readOnly
  fullAccess
  snapViewers

Volumes
-------
  /rcrds
    CIFS : compressed files: yes; named streams: yes; persistent ACLs: yes
           sparse files: yes; Unicode on disk: yes; case sensitive: no

             Volume freespace: 3.8 GB  automatic
           Volume total space: 5.9 GB
                  CIFS quotas: Not Enabled
              Auto Sync Files: Enabled
                Metadata size: 140 kB
          Metadata free space: 16 GB
              Filer Subshares: Enabled
               Oplock support: Enabled
           Notify-change mode: Normal
              CIFS path cache: Enabled
       CIFS access based enum: Not Enabled
                    Snapshots: Enabled
```

```
         Migration method: Staged
                    State: Enabled

              Host Switch: bstnA
                 Instance: 3
             Volume Group: 2
                Processor: 1.1
                    Files: 219 used (34 dirs), 3.9 M free, 248 M max (automatic)
Metadata shares:

  Filer          Backend Path           Contains Metadata  Status
  -------------------------------------------------------------------
  nas1           /vol/vol2/meta3        Yes                Online

Share bulk
  Description                 new server to hold big files (such as xrays)
  Filer                       fs2 [192.168.25.27]
  CIFS Share                  bulkstorage
  Features                    cifs-acls cifs-case-blind
  SID Translation             Yes
  Ignore SID errors           No
  Status                      Online
  Import Sync Attributes      Yes
  Import Priority             65535 (Lowest)
  Free space on storage       1.3 GB (1,408,800,768 B)
  Total space on storage      1.9 GB (2,138,540,032 B)
  Policy Maintain Freespace   5 %
  Policy Resume Freespace     6 %
  Transitions                 1
  Last Transition             Wed 07 Nov 2012 01:33:38 AM EST

Share charts
  Description                 various medical charts
  Filer                       fs1 [192.168.25.20]
  CIFS Share                  histories
  Features                    cifs-acls cifs-case-blind
  SID Translation             No
  Ignore SID errors           No
  Status                      Online
  Import Sync Attributes      Yes
  Import Priority             65535 (Lowest)
  Free space on storage       1.4 GB (1,515,354,112 B)
  Total space on storage      1.9 GB (2,138,540,032 B)
  Policy Maintain Freespace   5 %
  Policy Resume Freespace     6 %
  Transitions                 1
  Last Transition             Wed 07 Nov 2012 01:33:37 AM EST

Share rx
  Description                 prescriptions since 2002
  Filer                       fs4 [192.168.25.29]
  CIFS Share                  prescriptions
  Features                    cifs-acls cifs-case-blind
  SID Translation             No
  Ignore SID errors           No
  Status                      Online
  Volume Root Backing         Yes
  Critical Share              Yes
  Import Sync Attributes      Yes
  Import Priority             65535 (Lowest)
  Free space on storage       1.1 GB (1,225,224,192 B)
  Total space on storage      1.9 GB (2,144,333,824 B)
```

```
      Policy Maintain Freespace   5 %
      Policy Resume Freespace     6 %
      Transitions                 1
      Last Transition             Wed 07 Nov 2012 01:33:36 AM EST


bstnA# ...
```

Showing One Share

To show the configuration and status of one share in a managed volume, add the share clause after the volume clause. For example, the following command shows the configuration of the ‘wwmed~/acct~bills’ share:

```
bstnA# show namespace wwmed volume /acct share bills

Namespace "wwmed" Configuration
Description: namespace for World-Wide Medical network
Metadata Cache Size: 512 MB
NFS Character Encoding: ISO-8859-1

Supported Protocols
-------------------
  nfsv3 nfsv3-tcp

Participating Switches
----------------------
  bstnA (Volume Group 1) [Current Switch]


Volumes
-------
  /acct

             Volume freespace: 264 GB  automatic
           Volume total space: 328 GB
                Metadata size: 1.9 MB
          Metadata free space: 16 GB
                    Snapshots: Not Enabled
             Migration method: Staged
                        State: Enabled

                  Host Switch: bstnA
                     Instance: 2
                 Volume Group: 1
                    Processor: 1.1
                        Files: 4,433 used (439 dirs), 3.9 M free, 252 M max (automatic)
    Metadata shares:

      Filer          Backend Path           Contains Metadata  Status
      -------------------------------------------------------------------
      nas1           /vol/vol2/meta1        Yes                Online

    Share bills
      Filer                       das8 [192.168.25.25]
      NFS Export                  /work1/accting
      Features                    unix-perm
      Status                      Online
      Critical Share              Yes
      Import Sync Attributes      Yes
      Import Priority            65535 (Lowest)
      Free space on storage      23 GB (25,303,912,448 B)
      Total space on storage     70 GB (75,278,499,840 B)
      Policy Maintain Freespace  2 %
      Policy Resume Freespace    3 %
      Free files on storage      17M
      Transitions                1
      Last Transition            Wed 07 Nov 2012 01:32:53 AM EST


bstnA# ...
```

Showing Filer Shares Behind One Volume

You can use the show namespace mapping command to show the filer shares behind a particular namespace, as described earlier in the namespace chapter. Add the volume clause to show only the shares behind that particular volume; this is the same for managed volumes as described earlier for direct volumes (see *Showing Filer Shares Behind One Volume*, on page 8-28). For example, this shows the filer shares behind the “medarcv~/rcrds” volume:

```
bstnA# show namespace mapping medarcv volume /rcrds

Namespace               Physical Server
--------------------    ---------------------
medarcv:/rcrds
                        \\fs1\histories
                        \\fs2\bulkstorage
                        \\fs4\prescriptions
                        nas1:/vol/vol2/meta3*



Where * denotes metadata only physical server.
bstnA# ...
```

Showing the Volume’s Configuration

To review the configuration settings for a managed volume, identify the volume at the end of the show global-config namespace command. This is the same syntax used for showing direct-volume configuration; recall *Showing the Volume’s Configuration*, on page 8-29. The output shows all of the configuration options required to recreate the volume. The options are in order, so that they can be used as a CLI script.

For example, the following command shows the configuration for the “medarcv~/rcrds” volume:

```
bstnA# show global-config namespace medarcv /rcrds
;====================== namespace managed volumes ======================
namespace medarcv
  protocol cifs
  cifs authentication kerberos
  cifs authentication ntlm
  cifs authentication ntlmv2
  cifs filer-signatures
```

```
proxy-user acoProxy2
windows-mgmt-auth readOnly
windows-mgmt-auth fullAccess
windows-mgmt-auth snapViewers
sam-reference fs2
volume /rcrds
  filer-subshares
  modify
  policy migrate-method staged
  reimport-modify
  reserve files 4000000
  auto sync files
  metadata share nas1 nfs3 /vol/vol2/meta3
  compressed-files
  named-streams
  persistent-acls
  sparse-files
  unicode-on-disk
  no cifs path-cache
  share bulk
    description "new server to hold big files (such as xrays)"
    import sync-attributes
    policy freespace percent 5 resume-migrate 6
    sid-translation
    filer fs2 cifs bulkstorage
    enable
    exit

  share charts
    description "various medical charts"
    import sync-attributes
    policy freespace percent 5 resume-migrate 6
    filer fs1 cifs histories
    enable
    exit

  share rx
    description "prescriptions since 2002"
    import sync-attributes
    policy freespace percent 5 resume-migrate 6
    critical
    filer fs4 cifs prescriptions
    enable
    exit

  share-farm medFm
    share rx
    share charts
    balance latency
    enable
    exit

  snapshot rule rcrdsArchive
    schedule daily4am
    report FA_rcrds
    archive fileRecordsMed
    no contents user-data
    contents volume-config metadata
    enable
    exit

  place-rule dailyArchive
```

```
      schedule hourly
      report daily_archive
      from fileset onlineDayOld
      target share bulk
      no inline notify
      migrate close-file
      enable
      exit

    place-rule masterDirs2Rx
      from fileset allDirs match directories promote-directories
      target share rx
      enable
      exit

    volume-group 2
    enable
    exit

  exit

;======================= namespace direct volumes ========================
namespace medarcv
  protocol cifs
  cifs authentication kerberos
  cifs authentication ntlm
  cifs authentication ntlmv2
  cifs filer-signatures
  proxy-user acoProxy2
  windows-mgmt-auth readOnly
  windows-mgmt-auth fullAccess
  windows-mgmt-auth snapViewers
  sam-reference fs2
  exit

bstnA# ...
```

Sample - Configuring a Managed Volume

For example, this command set configures the ‘/acct’ volume on the ‘wwmed’ namespace. The ‘/acct’ volume contains two exports from two external filers:

```
bstnA(gbl)# namespace wwmed
This will create a new namespace.

Create namespace 'wwmed'? [yes/no] yes
bstnA(gbl-ns[wwmed])# volume /acct
This will create a new volume.

Create volume '/acct'? [yes/no] yes
bstnA(gbl-ns-vol[wwmed~/acct])# show external-filer
  Name                        IP Address     Description
  ------------------------    -------------  ----------------------------
  das1                        192.168.25.19  financial data (LINUX filer, rack 14)
  fs1                         192.168.25.20  misc patient records (Table 3)
  fs2                         192.168.25.27  bulk storage server (Table 3)
  fs3                         192.168.25.28  Hematology lab server (Table 8)
  fs4                         192.168.25.29  prescription records (Table 3)
  fs5                         192.168.25.71  docs, invoices, for scanners (Table 7)
  fs6                         192.168.25.30  records of lab animals - lab rats (rack C)
  fs7                         192.168.25.41  lab animal test results - lab rats (rack C)
  das2                        192.168.25.22  Solaris filer 2 (rack 16)
  das3                        192.168.25.23  Solaris filer 3 (rack 16)
  nas1                        192.168.25.21  NAS filer 1 (rack 31)
                              192.168.25.61   (1st secondary)
                              192.168.25.62   (2nd secondary)
  das7                        192.168.25.24  Redhat-LINUX filer 1
  das8                        192.168.25.25  Redhat-LINUX filer 2
  nas2                        192.168.25.44  NAS filer 2 (rack 31)
  nas3                        192.168.25.47  NAS filer 3 (rack 32)
  nas10                       192.168.25.49  NAS filer 10 (rack 38)
  nas11                       192.168.25.48  filer 11 (rack 38)
  nasE1                       192.168.25.51  NAS filer E1
  smb1                        192.168.25.48  Samba filer
bstnA(gbl-ns-vol[wwmed~/acct])# share budget
This will create a new share.

Create share 'budget'? [yes/no] yes
bstnA(gbl-ns-vol-shr[wwmed~/acct~budget])# filer das1 nfs3 /exports/budget
bstnA(gbl-ns-vol-shr[wwmed~/acct~budget])# enable
bstnA(gbl-ns-vol-shr[wwmed~/acct~budget])# exit
bstnA(gbl-ns-vol[wwmed~/acct])# share bills
This will create a new share.

Create share 'bills'? [yes/no] yes
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# filer das8 nfs3 /work1/accting
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# import sync-attributes
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# critical
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# enable
bstnA(gbl-ns-vol-shr[wwmed~/acct~bills])# exit
bstnA(gbl-ns-vol[wwmed~/acct])# enable
bstnA(gbl-ns-vol[wwmed~/acct])# exit
bstnA(gbl-ns[wwmed])# exit
bstnA(gbl)#
```

Removing a Managed Volume

As with a direct volume, use the priv-exec remove namespace ... volume command to remove a managed volume. (Recall *Removing a Direct Volume*, on page 8-31). For example, this command sequence exits to priv-exec mode and then removes the “medarcv~/lab_equipment” volume:

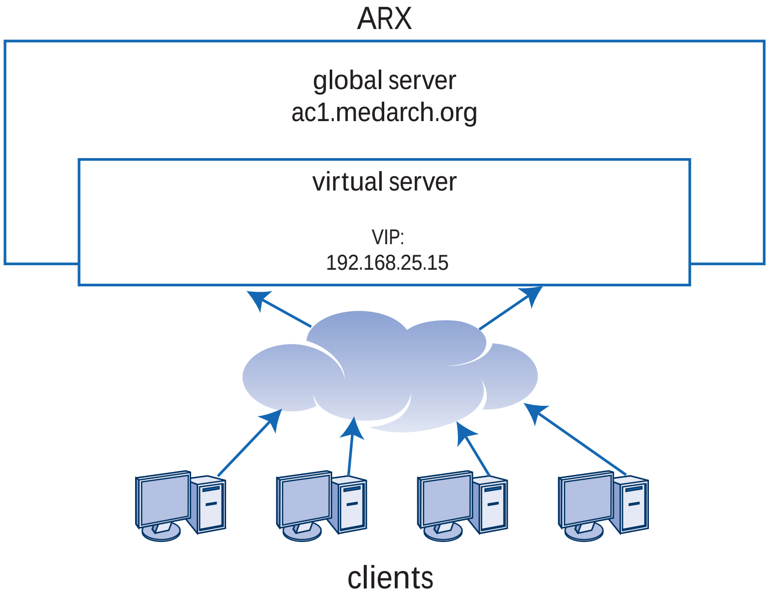
```
bstnA(gbl)# end
bstnA# remove namespace medarcv volume /lab_equipment

Remove volume '/lab_equipment' from namespace 'medarcv'? [yes/no] yes
Scheduling report: removeNs_medarcv_200702070957.rpt

bstnA# ...
```

10

Configuring a Global Server

Overview

A *global server* is a client-entry point to the ARX’s various front-end services. The global server defines a Fully-Qualified-Domain Name (FQDN; for example, “www.f5.com”) for accessing its services. A global server’s services are implemented by one *virtual server* on the ARX. Each virtual server listens at a unique virtual-IP (*VIP*) address.



Concepts and Terminology

A *front-end service* is a service that is visible to clients. This is in contrast to the *back-end* filers and servers, whose services are aggregated by the ARX. A front-end service provides an interface for clients to access the aggregated back-end services. For example, the NFS and CIFS front-end services provide mount points and share names for accessing various back-end filers.

A global-server configuration includes an FQDN. This FQDN is also used as the name for any of the global server’s front-end services. By convention, this is an FQDN used for client access.

Each global server contains a single *virtual server*. The virtual server is homed at a VIP address, which clients can use to access storage.

In a redundant pair, the global service and all of its components (the virtual server, its VIP, and any front-end services) fails over to the peer switch if the current switch ever fails.

Adding a Global Server

From gbl mode, use the global server command with a fully-qualified domain name (FQDN) to create a global server:

```
global server fqdn
```

where ***fqdn*** (1-128 characters) is the fully-qualified domain name (for example, www.company.com) to identify this service. By convention, this is the FQDN that clients will typically use to access the switch's resources.

The CLI prompts for confirmation before creating the global server; enter **yes** to proceed. This puts you into gbl-gs mode, where you must bind the server to one virtual server, set authentication parameters, and enable the global server.

For example, the following command sequence creates a global server, "ac1.medarch.org:"

```
bstnA(gbl)# global server ac1.medarch.org
This will create a new global server.

Create global server 'ac1.medarch.org'? [yes/no] yes
bstnA(gbl-gs[ac1.medarch.org])# ...
```

Setting the Windows Domain (CIFS Only)

If the global server uses back-end servers that require Windows authentication, the global server needs the Windows domain to integrate with the back-end servers. Use the windows-domain command to set the Windows domain:

```
windows-domain domain
```

where ***domain*** can be up to 64 characters long.

To support Kerberos authentications, this domain must be part of the Active-Directory (AD) forest that is known to the ARX. Recall *Discovering the Active-Directory Forest (Kerberos)*, on page 3-9.

For example, the following command sequence sets the domain to "MEDARCH.ORG" for the global server at "ac1.medarch.org:"

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# windows-domain MEDARCH.ORG
bstnA(gbl-gs[ac1.medarch.org])# ...
```

Setting the Pre-Windows2000 Name

Before the introduction of Windows 2000, Windows machines used short domain names, sometimes called "NT domain names," instead of the multi-domain strings in FQDNs. Today's Windows releases support both FQDNs and short-domain names. By default, the global server uses the first part of the global server's fully-qualified windows-domain name (up to 15 characters, converted to uppercase). For most installations, this is sufficient. For sites that do not conform to this naming convention, you can use the pre-win2k-name option to use a different short-domain name:

```
windows-domain domain pre-win2k-name short-name
```

where ***short-name*** (optional) can be up to 15 characters long. The CLI converts all letters to uppercase. As with the *domain*, the namespace behind this global server must have an NTLM-authentication server for the *short-name* domain, too. This is a separate NTLM-server configuration that points to the same server but supports the shortened Windows-domain name.

For example, the following command sequence sets the short name to "NTNET" for the global server at "ac1.medarch.org:"

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# windows-domain MEDARCH.ORG pre-win2k-name NTNET
bstnA(gbl-gs[ac1.medarch.org])# ...
```

Adding a Proxy User for Setting SPNs (Kerberos/AD Only)

This section applies to Active-Directory (AD) installations that use Kerberos authentication. You can skip this section for non-Kerberos site. You can also skip this section if this global server will not have any aliases, such as a DNS alias, a WINS name, or a WINS alias.

A global server with a CIFS service often uses multiple DNS or WINS aliases in addition to the server's FQDN. The AD keeps a database of these *service principal names* (SPNs) for all of its servers and clients. Each machine can have multiple SPNs. If a Kerberos client tries to connect to an unknown SPN, the Kerberos authentication fails. A later section explains how to register SPNs in the AD database.

The SPN registration will require proper Windows credentials at the local Domain Controller (DC); you provide those credentials by assigning a proxy user to the global server. (*Adding a Proxy User*, on page 3-4, described how to create a proxy-user object.) The proxy-user credentials must be valid in the global server's Windows domain, and must have sufficient privileges for setting SPNs. These privileges are stronger than those required for the backing namespace.

From gbl-gs mode, use the active-directory proxy-user command to assign a proxy user to the current global server:

```
active-directory proxy-user name
```

where ***name*** (1-32 characters) identifies the proxy user for this global server. Use the show proxy-user command for a list of configured proxy users.

For example, this command set applies a proxy user, "acoProxy2," to the "ac1.medarch.org" server:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# active-directory proxy-user acoProxy2
bstnA(gbl-gs[ac1.medarch.org])# ...
```

You can change the server's proxy user by re-running this command later and assigning a new one.

Removing the Windows Domain

Use no windows-domain to remove the Windows domain from the global server:

```
no windows-domain
```

For example:

```
bstnA(gbl)# global server www.nfs-only.com
bstnA(gbl-gs[www.nfs-only.com])# no windows-domain
bstnA(gbl-gs[www.nfs-only.com])# ...
```

Adding a Virtual Server

The next step in setting up a global server is creating one or more virtual servers. The virtual server listens at its VIP address for client requests; when one is received, the switch invokes one of the global server's front-end services (NFS or CIFS) to answer the client request. A global server requires one virtual server to run its front-end services.

Use the virtual server command to create a virtual server for an ARX and assign a VIP address to the switch:

```
virtual server switch-name virtual-ip-address mask [vlan vlan-id]
```

where

switch-name (1-128 characters) is the host name of the current ARX, and

virtual-ip-address is one VIP for the switch, which you create with this command.

mask establishes the subnet part of the virtual-IP address.

vlan *vlan-id* (optional; 1-65535) is the VLAN that carries the above subnet. The default is VLAN 1.

For example, suppose the local switch has the hostname, "bstnA." The following command sequence assigns it a VIP of 192.168.25.15 and adds the switch to the global server at "ac1.medarch.org:"

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[www.wwmed.com])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[www.wwmed.com~192.168.25.10])# ...
```

Registering with a WINS Server (CIFS)

This section only applies to virtual servers that support CIFS storage.

The Windows Internet Name Service (WINS) resolves domain-based names (such as www.mycompany.com) with IP addresses (such as 192.168.95.91). It is analogous to the Domain Name System (DNS), except that WINS integrates with the dynamic-addressing protocol, Dynamic Host Configuration Protocol (DHCP).

If you identify a WINS server for this network, the virtual server registers its NetBIOS name with the WINS server. This makes it possible for other WINS clients to find the virtual server on this switch. Use the wins command to identify a WINS server:

```
wins ip-address
```

where ***ip-address*** is the address of the name server.

If the WINS server supports multi-byte character encoding, set the proper character encoding at the namespace behind this virtual server. Refer back to *Returning to Default Character Encoding For NFS*, on page 7-12.

For example, this command sequence configures a WINS server for the virtual server at VIP 192.168.25.15:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# wins 192.168.25.20
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# ...
```

Removing the WINS-Server Setting

You can stop the virtual server from registering its NetBIOS name with a WINS server. This is only effective if you do it before you enable the virtual server (below). Use the no form of the wins command to remove the IP address of the WINS server:

```
no wins
```

◆ **Note**

The virtual server answers broadcast requests for its shares, whether it is registered with a WINS server or not.

For example:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# no wins
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# ...
```

Setting the NetBIOS Name (optional, CIFS)

This section only applies to virtual servers that support CIFS storage.

The virtual server’s NetBIOS name is the server name that appears in Windows network browsers. This appears in the “Server Name” column when you issue a Windows **net view** command. For example:

```
U:\>net view
Server Name            Remark
------------------------------------------------------------------------------
\\PERSONNEL
\\ARCHIVE1         A full tree of 5-year-old records
```

The above net view displays two CIFS servers, PERSONNEL and ARCHIVE1.

The default NetBIOS name for the virtual server is the first component of the global server's FQDN; for example, "FS" for the global server at "fs.nt.org." If that component is longer than 15 bytes, only the first 15 bytes are used.

Use the optional wins-name command to reset the name:

```
wins-name netbios-name
```

where ***netbios-name*** (1-15 bytes) is the NetBIOS name to be advertised to the WINS server. The first character must be a letter, and the remaining characters can be letters, numbers, or underscores (_). If you are using Latin characters (including ASCII), each character is one byte, so the name can be up to 15 characters. Each character may be more than one byte for non-Latin characters, limiting you to a smaller number of characters.

For example, this command sequence resets the NetBIOS name to "INSURANCE" for the virtual server at 192.168.25.15:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# wins-name INSURANCE
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# ...
```

Adding a NetBIOS Alias

Some installations use multiple NetBIOS names for a single CIFS server. To mimic this configuration, use the wins-alias command (in gbl-gs-vs mode) for each additional NetBIOS name:

```
wins-alias netbios-alias
```

where ***netbios-alias*** (1-15 bytes) is a NetBIOS alias to be advertised to the WINS server. The first character must be a letter, and the remaining characters can be letters, numbers, or underscores (_). The same character/byte limitations apply to this name as the NetBIOS name itself (see above).

For example, this command sequence adds a NetBIOS alias to the "192.168.25.12" server:

```
bstnA(gbl)# global server fs.nt.org
bstnA(gbl-gs[fs.nt.org])# virtual server bstnA 192.168.25.12 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[fs.nt.org~192.168.25.12])# wins-alias HR_DISK
bstnA(gbl-gs-vs[fs.nt.org~192.168.25.12])# ...
```

Removing NetBIOS Aliases

To remove one NetBIOS alias (or all of them) from the virtual server, use no wins-alias:

```
no wins-alias [netbios-name]
```

where ***netbios-name*** (optional, 1-15 characters) identifies the alias to remove. If you do not specify a particular NetBIOS alias, this command removes all of them.

The CLI prompts for confirmation if you choose to remove all aliases. Enter **yes** to continue.

For example, this command sequence removes all NetBIOS aliases from the virtual server at 192.168.25.15:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# no wins-alias
Delete all NETBIOS aliases for this Virtual Server? [yes/no] yes
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# ...
```

Reverting to the Default NetBIOS Name

You can revert to the default NetBIOS name with the no wins-name command. The default NetBIOS name is the first component of the global server's FQDN (for example, "\\FTP1" for global server "ftp1.government.gov").

no wins-name

The CLI prompts for confirmation before deleting the name.

For example, the following command sequence sets the NetBIOS name back to its default ("\\MYCO") for the virtual server at "192.168.25.87:"

```
bstnA(gbl)# global server myco.com
bstnA(gbl-gs[myco.com])# virtual server bstnA 192.168.25.87 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[myco.com~192.168.25.87])# no wins-name
Delete all NETBIOS aliases for this Virtual Server? [yes/no] yes
bstnA(gbl-gs-vs[myco.com~192.168.25.87])# ...
```

Setting SPNs (CIFS/Kerberos Only)

This section only applies to virtual servers that support CIFS storage in an Active-Directory (AD) environment. It does not apply to an installation that only uses NTLM and/or NTLMv2 authentication. You can also skip this section if you do not plan for any DNS aliases, WINS names, or WINS aliases for this virtual server.

When the virtual server's CIFS service joins its Windows domain, it registers the global server's FQDN name in the AD database. This FQDN becomes the *service principal name* (SPN) for the service: the DCs use this name to identify the virtual server for Kerberos authentications. If a client uses an alias name to connect to a CIFS service, a DC cannot use WINS or DNS to translate the alias to the actual SPN. This causes Kerberos authentication to fail.

You may have already registered some aliases through wins-name and/or wins-alias, described above. You may also have some DNS aliases for this virtual server. A CLI command can register these aliases as SPNs.

◆ **Note**

*Before you set any SPNs, the global server requires a proxy user with adequate credentials for doing so. Recall **Adding a Proxy User for Setting SPNs (Kerberos/AD Only)**, on page 10-5.*

From gbl-gs-vs mode, use the active-directory alias command to register one SPN for each of the server's aliases:

active-directory alias *spn*

where *spn* (1-256 characters) is one SPN for the current virtual server.

This sets a "HOST" SPN as well as a "CIFS" SPN at the local DC. The local DC is the currently-active DC in the global server's Windows Domain; use the show active-directory command for a list of all Windows Domains and their corresponding DCs. *Showing All Active-Directory Forests*, on page 3-16, describes the show active-directory command. The virtual server sets the SPN as soon as it and its global server are both enabled (described below).

For example, the following command sequence sets 8 SPNs for the virtual server at "192.168.25.15:"

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory alias fs1
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory alias fs2
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory alias fs5
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory alias fs9
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory alias fs1.medarch.org
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory alias fs2.medarch.org
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory alias fs5.medarch.org
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory alias fs9.medarch.org
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# ...
```

With these SPN mappings, a client can use Kerberos authentication if he or she connects with any of these aliases. Use DNS to ensure that these aliases map to the current server.

Removing a SPN

To remove a SPN from the AD database, use the no active-directory alias command:

```
no active-directory alias [spn]
```

where *spn* (optional, 1-256 characters) is a single SPN to remove for this virtual server. If you omit this option, the command removes all of the virtual server's SPNs from the AD database. The CLI prompts for confirmation before removing all of the server's SPNs.

As with the affirmative form of the command, this only removes the SPN(s) when both the virtual server and the global server are enabled.

For example, this command sequence removes the "fs9" SPNs from the above virtual server:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# no active-directory alias fs9
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# no active-directory alias fs9.medarch.org
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# ...
```

Enabling a Virtual Server

The final step in virtual-server configuration is to enable it. You must explicitly enable a virtual server for the switch to accept clients at its VIP address. From gbl-gs-vs mode, use the enable command to activate the virtual server:

```
enable
```

This makes it possible for the virtual server to accept clients. (You must also enable the global server, so that it offers front-end services to those clients.)

For example, the following command sequence enables the virtual server on "bstnA" at 192.168.25.15, and then enables its global server at ac1.medarch.org:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# enable
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# exit
bstnA(gbl-gs[ac1.medarch.org])# enable
bstnA(gbl-gs[ac1.medarch.org])# ...
```

Disabling a Virtual Server

Disabling a virtual server makes it impossible for clients to access the particular switch's front-end services (such as CIFS or NFS) through that virtual server's IP address. Use no enable in gbl-gs-vs mode to disable a virtual server.

```
no enable
```

This command gracefully terminates all client connections at the VIP address, allowing current transactions and sessions to finish while blocking any new connections.

For example, the following command sequence disables a virtual server:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# no enable
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# ...
```

Removing a Virtual Server

Once you remove a virtual server, the global server's front-end services stop running on the virtual server's ARX.

Use the no form of the virtual server command to remove a virtual server:

```
no virtual server switch-name virtual-ip-address
```

where

switch-name (1-128 characters) is the host name of the ARX, and

virtual-ip-address identifies the virtual server.

Before removing the virtual server, the CLI prompts for confirmation. Enter **yes** to continue.

For example, the following command sequence removes a virtual server from global server ac1.medarch.org:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# no virtual server bstnA 192.168.123.9
Delete virtual server ''192.168.123.9'' on switch ''bstnA''? [yes/no] yes
bstnA(gbl-gs[ac1.medarch.org])# ...
```

Enabling the Global Server

The final step in global-server configuration is to enable it. Use the enable command to activate the global server:

```
enable
```

For example, the following command sequence enables the global server at "ac1.medarch.org:"

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# enable
bstnA(gbl-gs[ac1.medarch.org])# ...
```

Disabling the Global Server

Disabling the global server makes it impossible for clients to access any of its resources. Use no enable in gbl-gs mode to disable the global server.

```
no enable
```

This command gracefully terminates all client connections to the global server, allowing current transactions and sessions to finish while blocking any new connections.

For example, the following command sequence disables the "ac1.medarch.org" global server:

```
bstnA(gbl)# global server ac1.medarch.org
bstnA(gbl-gs[ac1.medarch.org])# no enable
bstnA(gbl-gs[ac1.medarch.org])# ...
```

Showing All Global Servers

Use the show global server command to show all global servers on the ARX:

```
show global server
```

For example:

```
bstnA# show global server

Domain Name             State                Windows Domain
---------------------------------------------------------------------------
ac1.MEDARCH.ORG            Enabled              MEDARCH.ORG

  Switch                State    VIP                VLAN  VMAC
  -------------------------------------------------------------------------
  bstnA                 Enabled  192.168.25.15/24    25   00:0a:49:17:92:c0

  Aliases:

    SPNs
      Name                                               AD Status
      --------------------------------------------------   ----------------
      CIFS/insur                                                OK
      HOST/insur                                                OK
      CIFS/fs1                                                  OK
      HOST/fs1                                                  OK
      CIFS/fs1.MEDARCH.ORG                                      OK
      HOST/fs1.MEDARCH.ORG                                      OK
      CIFS/fs2                                                    OK
      HOST/fs2                                                    OK
      CIFS/fs2.MEDARCH.ORG                                        OK
      HOST/fs2.MEDARCH.ORG                                        OK
      CIFS/fs5                                                    OK
      HOST/fs5                                                    OK
      CIFS/fs5.MEDARCH.ORG                                        OK
      HOST/fs5.MEDARCH.ORG                                        OK
      CIFS/insur                                                  OK
      HOST/insur                                                  OK
      CIFS/insur.MEDARCH.ORG                                      OK
      HOST/insur.MEDARCH.ORG                                      OK

    WINS
      Name                                                 WINS Server IP
      ---------------------------------------------------------------------
      INSURANCE                                            192.168.25.102


  Description
  -------------------------------------------------------------------------
  CIFS and NFS server for hospital insurance claims

Domain Name             State                Windows Domain
---------------------------------------------------------------------------
acopiaFiler             Enabled

  Switch                State    VIP               VLAN  VMAC
  -------------------------------------------------------------------------
  bstnA                 Enabled  192.168.25.12/24    25   00:0a:49:17:92:c0

  Aliases:

    SPNs      No SPNs are defined.
    WINS
      Name                                                 WINS Server IP
      ---------------------------------------------------------------------
      ACOPIAFILER                                          (none)


  Description
  -------------------------------------------------------------------------


bstnA#
```

Showing One Global-Server

To see one global server, identify a particular global server with the show global server command:

show global server *fqdn*

where ***fqdn*** (1-128 characters) is the fully-qualified domain name (for example, www.company.com) for the global server.

For example, the following command shows the global server at “www.nemed.com:”

```
bstnA# show global server www.nemed.com

Domain Name             State                Windows Domain
```

```
---------------------------------------------------------------------------
www.nemed.com           Enabled              MEDARCH.ORG

  Switch                State    VIP                VLAN  VMAC
  -------------------------------------------------------------------------
  bstnA               Enabled  192.168.74.91/24     74  00:0a:49:17:b9:c0

  Aliases:

    SPNs      No SPNs are defined.
    WINS
      Name                                                 WINS Server IP
      ---------------------------------------------------------------------
      WWW                                                  (none)


  Description
  -------------------------------------------------------------------------
  global NFS server for network hospitals
bstnA# ...
```

Removing a Global Server

You cannot remove the global server if any front-end services reference it, or if it is enabled. After you remove all services and disable the global server, use the no form of the global server command to remove it:

`no global server` ***`fqdn`***

where ***fqdn*** (1-128 characters) is the fully-qualified domain name (for example, www.company.com) for the global server.

The CLI prompts for confirmation before deleting the global server; enter **yes** to continue. For example, the following command sequence removes global server ftp.medarch.org:

```
bstnA(gbl)# global server ftp.medarch.org
bstnA(gbl-gs[ftp.medarch.org])# no enable
bstnA(gbl-gs[ftp.medarch.org])# exit
bstnA(gbl)# no global server ftp.medarch.org
Delete global server 'ftp.medarch.org'? [yes/no] yes
bstnA(gbl)# ...
```

Sample - Configuring a Global Server

The following command sequence sets up a global server for ac1.medarch.org.

Create the global server:

```
bstnA(gbl)# global server ac1.medarch.org
This will create a new global server.

Create global server 'ac1.medarch.org'? [yes/no] yes
```

Join a Windows domain, MEDARCH.ORG:

```
bstnA(gbl-gs[ac1.medarch.org])# windows-domain MEDARCH.ORG
```

Assign a proxy user with strong enough privileges to set SPNs for the server:

```
bstnA(gbl-gs[ac1.medarch.org])# active-directory proxy-user acoProxy2
```

Bind to the current ARX, “bstnA.” Assign a VIP address, 192.168.25.15, to the server:

```
bstnA(gbl-gs[ac1.medarch.org])# virtual server bstnA 192.168.25.15 255.255.255.0 vlan 25
```

Identify the local WINS server:

```
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# wins 192.168.25.20
```

Set some SPNs for the virtual server, so that Kerberos clients can use these names successfully:

```
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory fs1
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory fs2
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory fs5
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory fs1.medarch.org
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory fs2.medarch.org
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# active-directory fs5.medarch.org
```

Enable and exit both the virtual server and the global server:

```
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# enable
bstnA(gbl-gs-vs[ac1.medarch.org~192.168.25.15])# exit
bstnA(gbl-gs[ac1.medarch.org])# enable
bstnA(gbl-gs[ac1.medarch.org])# exit
bstnA(gbl)#
```

Next

For a global server to host any back-end services (such as filer storage), you must configure one or more front-end services. The front-end services provide access to namespace storage. *Chapter 11, Configuring Front-End Services*, describes how to configure various front-end services for a global server.

11

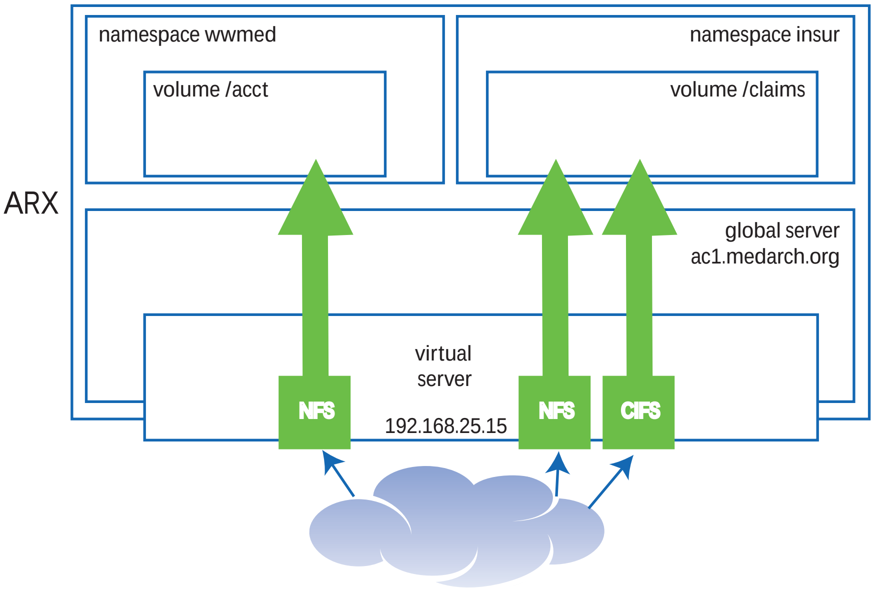
Configuring Front-End Services

Overview

Front-end services provide client access to namespace storage. Supported front-end services include:

- Network File System (NFS), and
- Common Internet File System (CIFS).

You can enable one or more of these services on a global server, so that clients can access them through the global server's fully-qualified domain name (FQDN) or the virtual server's VIP. For each service, you determine which namespace volumes are available as storage resources.



Before You Begin

To offer any front-end services, you must first

- add one or more NAS filers, as described in *Chapter 6, Adding an External Filer*;
- create at least one namespace as a storage resource, as described in *Chapter 7, Configuring a Namespace*;
- create at least one volume in the namespace, a direct volume (recall *Chapter 8, Adding a Direct Volume*) or a managed volume (see *Chapter 9, Adding a Managed Volume*);

  and
- configure a global server that uses this ARX as a virtual server, as described in *Chapter 10, Configuring a Global Server*.

Configuring NFS

From gbl mode, use the nfs command to instantiate NFS service for a global server:

```
nfs fqdn
```

where ***fqdn*** (1-128 characters) is the fully-qualified domain name (for example, "www.organization.org") for the global server.

The CLI prompts for confirmation before creating the service; enter **yes** to proceed. (You can disable all such create-confirmation prompts with the terminal expert command.) This places you in gbl-nfs mode, from which you can export various namespace volumes.

For example, the following command sequence creates an NFS service at ac1.medarch.org:

```
bstnA(gbl)# nfs ac1.medarch.org
This will create a new NFS service.

Create NFS service 'ac1.medarch.org'? [yes/no] yes
bstnA(gbl-nfs[ac1.medarch.org])# ...
```

From gbl-nfs mode, you must export at least one namespace volume and then enable the NFS service, as described in the following subsections.

Exporting a Namespace Volume

If a namespace volume is configured for NFS, you can offer it as an NFS export through a global server. Each NFS service can support volumes from one or more namespaces.

Use the export command to offer a namespace volume through the current NFS service:

```
export namespace vol-path [as export-path] [access-list list-name]
```

where

namespace (1-30 characters) can be any namespace that supports NFS,

vol-path (1-1024 characters) is the path to one of the namespace's volumes (for example, "/etc") or volume sub paths ("/etc/init.d"), and

as *export-path* (optional, 1-1024 characters) sets an optional, advertised path to the volume. If entered, NFS clients see this path as an available export instead of the ***volume*** path.

access-list *list-name* (optional, 1-64 characters) uses an access list to control which IP subnets can and cannot access this NFS service. See *Adding an NFS Access List*, on page 4-9 for instructions on configuring an NFS access list. If you omit this option, clients from *any* subnet can access the export with read/write permission, root-squash enabled, and root is squashed to UID 65534 and GID 65534.

Whether or not you use an access list, non-root UIDs are passed through to the back-end filer for possible authentication at the filer.

Use show global-config namespace to see all available namespaces and volumes on the ARX. Each NFS service can support volumes from *one* namespace only. Choose only volumes that support NFS.

For example, the following command sequence adds an NFS export to the NFS service at ac1.medarch.org:

```
bstnA(gbl)# nfs ac1.medarch.org
bstnA(gbl-nfs[www.wwmed.com])# show global-config namespace wwmed
;====================== namespace managed volumes =======================
namespace wwmed
  protocol nfs3
  protocol nfs3tcp
  description "namespace for World-Wide Medical network"
  volume /acct
    metadata critical
    modify
    policy migrate-method staged
    policy pause backupWindow
    reimport-modify
    reserve files 4000000
    metadata share nas1 nfs3 /vol/vol2/meta1
    share bills
      import sync-attributes
      policy freespace percent 2 resume-migrate 3
      critical
      filer das8 nfs /work1/accting
      enable
      exit

    share bills2
      import sync-attributes
      policy freespace percent 2 resume-migrate 3
      filer das3 nfs /exports/acct2
      enable
      exit

    share budget
      policy freespace percent 2 resume-migrate 3
      filer das1 nfs /exports/budget
      enable
      exit

    share it5
      import sync-attributes
      policy freespace percent 2 resume-migrate 3
      filer das7 nfs /lhome/it5
      enable
      exit

    share-farm fm1
      share bills
      share budget
      share bills2
      balance capacity
      auto-migrate
      enable
      exit
    place-rule docs2das8
      report docsPlc verbose
      inline report hourly docsPlc verbose
      from fileset bulky
      target share bills
      limit-migrate 50G
      enable
      exit

    volume-group 1
    enable
    exit

  exit

bstnA(gbl-nfs[www.wwmed.com])# export wwmed /acct access-list eastcoast
bstnA(gbl-nfs[www.wwmed.com])# ...
```

Stopping an NFS Export

Use the no form of the export command to stop NFS access to a namespace volume:

no export *namespace vol-path*

where

namespace (1-30 characters) is the namespace containing the NFS export, and

vol-path (1-1024 characters) is the path to a current NFS export (for example, “/acct”).

For example, the following command sequence stops the NFS server at www.wwmed.com from exporting “wwmed /test”:

```
bstnA(gbl)# nfs www.wwmed.com
bstnA(gbl-nfs[www.wwmed.com])# no export wwmed /test
bstnA(gbl-nfs[www.wwmed.com])# ...
```

Disabling NLM (optional)

The NFS service implements the NFS Lock Manager (NLM) protocol. NLM is a voluntary protocol that NFS-client applications can use to write-protect a file or file region. NFS client A can use NLM to *lock* a region of a file; if clients B and C are also NLM-compliant, they will not write to that region until client A releases the lock.

While the NFS service is disabled, you have the option to disable NLM.

◆ **Note**

This requires careful consideration for an NFS service in front of a direct volume (direct volumes were described in Chapter 8, Adding a Direct Volume). A direct volume can offer NLM file locks to its clients, but clients that access the back-end filers directly (not through the ARX) or through *another direct volume are unaware of those locks. Therefore, direct volumes should not offer NLM unless you are sure that all client access goes through this volume.*

If multiple NFS services export the same volume, consistently enable or disable NLM for all of them. This applies to managed volumes as well as direct volumes.

Use the no nlm enable command from gbl-nfs mode to disable NLM at this NFS service:

```
no nlm enable
```

This prevents the front-end service from answering any NLM requests for file locks; the CLI prompts for confirmation before doing this. Enter **yes** to continue.

For example:

```
bstnA(gbl)# nfs acopiaFiler
bstnA(gbl-nfs[acopiaFiler])# no nlm enable
Disable Network Lock Manager on acopiaFiler? [yes/no] yes
bstnA(gbl-nfs[acopiaFiler])# ...
```

For commands to view current NLM locks and NLM statistics, refer to the Front-End Services chapter in the *ARX® CLI Reference*.

Enabling NLM

While the NFS service is disabled, you can use nlm enable to re-enable NLM processing:

```
nlm enable
```

This causes the front-end service to answer all NLM requests.

For example:

```
bstnA(gbl)# nfs www.wwmed.com
bstnA(gbl-nfs[www.wwmed.com])# nlm enable
bstnA(gbl-nfs[www.wwmed.com])# ...
```

Enabling NFS Service

The final step in NFS configuration is to enable it. Use the enable command from gbl-nfs mode to activate the NFS service:

```
enable
```

This makes all declared NFS exports available to clients at the global server’s VIP.

For example, the following command sequence enables NFS for the global server at “www.wwmed.com:”

```
bstnA(gbl)# nfs www.wwmed.com
bstnA(gbl-nfs[www.wwmed.com])# enable
bstnA(gbl-nfs[www.wwmed.com])# ...
```

Disabling NFS

Disabling NFS stops all NFS connections to the global server. Use no enable in gbl-nfs mode to disable NFS.

```
no enable
```

For example, the following command sequence disables NFS for the "www.wwmed.com" global server:

```
bstnA(gbl)# nfs www.wwmed.com
bstnA(gbl-nfs[www.wwmed.com])# no enable
bstnA(gbl-nfs[www.wwmed.com])# ...
```

Notifications to NLM Clients

As described above, the NFS service can implement the NFS Lock Manager (NLM) protocol. If you used no nlm enable to stop NLM, skip to the next section.

In an NFS service where NLM is enabled, a no enable followed by an enable triggers a notification to NLM clients. When the NFS service comes back up, it follows the NLM protocol for server recovery from a crash. The NFS service sends an SM_NOTIFY message to all NLM clients with file locks. According to the protocol, all clients should then reclaim their locks.

For commands to view current NLM locks and NLM statistics, refer to the Front-End Services chapter in the *ARX® CLI Reference*.

Listing All NFS Services

Use the show nfs-service command to see summaries for all NFS services:

```
show nfs-service
```

For example:

```
bstnA(gbl)# show nfs-service

Domain Name                Description
-------------------------------------------------
acopiaFiler
ac1.MEDARCH.ORG            insurance records for WW Medical

bstnA(gbl)# ...
```

Showing One NFS Service

Identify a particular FQDN with the show nfs-service command to focus on one NFS service:

```
show nfs-service fqdn
```

where ***fqdn*** (1-128 characters) is the fully-qualified domain name (for example, www.company.com) for the global server.

This shows detailed configuration information for the service.

For example, the following command shows the NFS configuration for the global server at “ac1.MEDARCH.ORG:”

```
bstnA(gbl)# show nfs-service ac1.MEDARCH.ORG

Virtual Server: ac1.MEDARCH.ORG
Description:    insurance records for WW Medical

NFS State:      Enabled
NLM State:      Enabled

Exports for Namespace: insur

  Export As            Volume Path               State       Access
  -----------------------------------------------------------------------
  /claims              /claims                   Online      eastcoast (NIS 192.168.25.2)

Exports for Namespace: wwmed

  Export As            Volume Path               State       Access
  -----------------------------------------------------------------------
  /acct                /acct                     Online      eastcoast (NIS 192.168.25.2)

bstnA(gbl)#
```

Showing Details for All NFS Services

To show details for all NFS front-end services, use show nfs-service all:

```
show nfs-service all
```

This shows the detailed view of all configured NFS services.

Sample - Configuring an NFS Front-End Service

The following command sequence sets up NFS service on a global server called “www.wwmed.com:”

```
bstnA(gbl)# nfs www.wwmed.com
bstnA(gbl-nfs[www.wwmed.com])# show global-config namespace wwmed
;====================== namespace managed volumes ======================
namespace wwmed
  protocol nfs3
  protocol nfs3tcp
  description "namespace for World-Wide Medical network"
  volume /acct
    metadata critical
    modify
    policy migrate-method staged
    policy pause backupWindow
    reimport-modify
    reserve files 4000000
    metadata share nas1 nfs3 /vol/vol2/meta1
    share bills
      import sync-attributes
      policy freespace percent 2 resume-migrate 3

    ...
bstnA(gbl-nfs[www.wwmed.com])# export wwmed /acct access-list eastcoast
bstnA(gbl-nfs[www.wwmed.com])# enable
bstnA(gbl-nfs[www.wwmed.com])# exit
bstnA(gbl)# show nfs-service www.wwmed.com

Virtual Server: www.wwmed.com
Namespace:      wwmed
Description:

NFS State:      Enabled
NLM State:      Enabled

Exports:

  Directory             Export As               State      Access
  ------------------------------------------------------------------------
  /acct                 /acct                   Online     eastcoast
bstnA(gbl)# ...
```

Removing an NFS Service

You can remove an NFS service from a global server to both disable the service and remove its configuration. Use the no form of the nfs command to remove an NFS-service configuration from a global server:

no nfs *fqdn*

> where ***fqdn*** (1-128 characters) is the fully-qualified domain name (for example, “www.organization.org”) for the service’s global server.

The CLI prompts for confirmation before removing the service; enter **yes** to proceed.

For example, the following command sequence removes the NFS-service offering for athos.nfstest.net:

```
bstnA(gbl)# no nfs athos.nfstest.net
Delete NFS service on 'athos.nfstest.net'? [yes/no] yes
bstnA(gbl)#
```

Changing the NFS/TCP Timeout Behavior

When an NFS/TCP connection to a back-end share times out, the NFS service (by default) disconnects its front-end client. You can change this default: instead of disconnecting from the client, the NFS service can send back an NFS I/O error (NFSERR_IO or NFS3ERR_IO). With the same command, you can reduce the default time-out period of 105 seconds. From gbl mode, use the nfs tcp timeout command to make these changes:

nfs tcp timeout *seconds*

> where ***seconds*** (5-104) is the time-out period for NFS/TCP connections.

For example, this command sets the timeout to 30 seconds (and stops disconnecting clients on timeout):

```
bstnA(gbl)# nfs tcp timeout 30
bstnA(gbl)# ...
```

Showing the NFS/TCP Timeout

Use the show nfs tcp command to view the current client-connection behavior and timeout period for NFS/TCP timeouts:

```
show nfs tcp
```

For example, this system has the behavior configured above:

```
bstnA(gbl)# show nfs tcp

Transaction Timeout
  Behavior:    Return I/O Error
  Inactivity:  30 seconds
bstnA(gbl)# ...
```

Reverting to the Default Timeout and Behavior

By default, an NFS/TCP connection to any filer times out in 105 seconds, and the NFS service drops the connection to the client on timeout. Use no nfs tcp timeout to return to this default for all NFS filers and services:

```
no nfs tcp timeout
```

For example:

```
bstnA(gbl)# no nfs tcp timeout
bstnA(gbl)# ...
```

Controlling Access to Offline Shares

By default, when an NFS client attempts to access an NFS export that is not available, the client keeps retrying the request and waiting for that export to respond, which may result in the request hanging indefinitely. This occurs when an NFS filesystem goes offline, and can be problematic when an NFS service exported by the ARX comprises multiple NFS filesystems. Even though the other NFS file systems are online and accessible, an NFS request, such as a simple ls -l command, will hang once it encounters the offline filesystem.

The gbl-nfs mode CLI command, offline-behavior, provides an optional means of avoiding delays caused by NFS servers that are unable to respond to requests: offline-behavior deny-access allows offline NFS exports to return an access error, typically, either NFSERR_ACCES or NFS3ERR_ACCES, which results in the NFS client indicating the export with “Permission denied”. In this case, the NFS request continues operation, accessing those NFS exports that are online, without the request hanging while it awaits a response from the offline NFS filesystem.

The default behavior is manifested by the CLI command, offline-behavior retry, which applies the retry behavior to all namespaces in the NFS service.

The syntax for the offline-behavior command is:

```
offline-behavior [namespace ns [volume /vol [path expt-path]]]
  { retry | deny-access }
```

where:

ns (optional, 1-30 characters) narrows the scope of this command to a particular namespace behind this NFS service.

/vol (optional if you choose namespace, 1-1024 characters) narrows the scope further, to a particular volume in the above namespace.

expt-path (optional, if you choose a volume, 1-64 characters) narrows the scope to one explicitly-exported path in the volume. You must choose an export path that is explicitly exported with the export (gbl-nfs) command. If this path is not exported explicitly, the command is accepted, but the NFS service does not exhibit the offline behavior until someone exports this exact path.

retry | deny-access is a required choice:

retry causes the service to accept requests to offline filers without sending an error back to the NFS client. The client must therefore retry until the filer comes back online.

deny-access causes the service to return an error (typically NFSERR_ACCES or NFS3ERR_ACCES) for any client request that requires an offline filer.

For example, executing this command:

```
bstnA(gbl-nfs[ac1.medarch.org])# offline-behavior deny-access
```

causes the "ac1.medarch.org" service to return an access error to NFS clients that access an offline filer, rather than retrying a request indefinitely.

Specifying Exceptions to the Namespace's Overall Offline Behavior

Using the offline-behavior command's optional volume and path arguments, you can specify one or more exceptions to the overall offline behavior that is configured for a namespace. For example, for a namespace that is configured with offline-behavior deny-access, you can specify one or more exports in that namespace to which you want to restore the "retry" behavior as an exception to the overall behavior.

For example, this command:

```
bstnA(gbl-nfs[ac1.medarch.org])# offline-behavior namespace wwmed
volume /acct path /wksheets retry

% WARNING: This command did not match any existing export. It will
have no effect until an export matching '/acct/wksheets' is entered in
this NFS service.
```

overrides the default behavior for clients that connect to the /acct/wksheets export. The warning shown here indicates that the export does not yet exist, so NFS clients that mount /acct/wksheets still get the access errors for offline filers. You can use the export (gbl-nfs) command to create the /acct/wksheets export later, and that export will exhibit the retry behavior once it has been created.

Using the “no offline-behavior” Command

Use the no offline-behavior command to remove an offline-behavior command that had been specified previously.

The syntax for the no offline-behavior command follows:

```
no offline-behavior [namespace ns [volume /vol
  [path expt-path]]]
```

For example, this command:

```
bstnA(gbl-nfs[ac1.medarch.org])# no offline-behavior namespace insur
```

disables the “deny-access” behavior for exports from the “insur” namespace. This overrides the default set for the NFS service.

The no offline-behavior command tries to match an offline-behavior command that specifies the same namespace/volume/path parameters, removes it if it finds it, and does nothing if it does not find it.

For example, if you execute this:

```
offline-behavior namespace eng volume /builds path /usr/local/bin
deny-access
```

and subsequently execute this:

```
no offline-behavior namespace eng volume /builds path /usr/local/bin
retry
```

It will remove the command:

```
offline-behavior namespace eng volume /builds path /usr/local/bin
deny-access
```

because it matches the namespace/volume/path, returning the export to the default “retry” behavior.

Executing the no offline-behavior command without a namespace/volume/path will remove a command that specified offline-behavior deny-access without a namespace/volume/path, if one had been entered. It has no effect on any other offline-behavior commands, however.

For example, suppose you have the following NFS service:

```
bstnA(gbl-nfs[ac1.MEDARCH.ORG])# show global-config nfs ac1.MEDARCH.ORG
;================================== nfs ==================================
nfs ac1.MEDARCH.ORG
  description "insurance records for WW Medical"
  export wwmed /acct as /acct access-list eastcoast
  export wwmed /acct/wksheets as /acct/wksheets access-list eastcoast
  export insur /claims as /claims access-list eastcoast
  offline-behavior namespace insur deny-access
  offline-behavior namespace wwmed volume /acct path wksheets deny-access
  enable
  exit
```

When you do this, you see that the no offline-behavior command has no effect:

```
bstnA(gbl-nfs[ac1.MEDARCH.ORG])# no offline-behavior
bstnA(gbl-nfs[ac1.MEDARCH.ORG])# show global-config nfs ac1.MEDARCH.ORG
;================================== nfs ==================================
nfs ac1.MEDARCH.ORG
  description "insurance records for WW Medical"
  export wwmed /acct as /acct access-list eastcoast
  export wwmed /acct/wksheets as /acct/wksheets access-list eastcoast
  export insur /claims as /claims access-list eastcoast
  offline-behavior namespace insur deny-access
  offline-behavior namespace wwmed volume /acct path wksheets deny-access
  enable
  exit
```

The no offline-behavior command would have removed an offline-behavior deny-access command, if one had been entered. However, it has no effect on those commands that are present and that specify a namespace/volume/path.

Configuring CIFS

From gbl mode, use the cifs command to instantiate CIFS service for a global server:

```
cifs fqdn
```

where ***fqdn*** (1-128 characters) is the fully-qualified domain name for the global server (for example, “myserver.organization.org”). If this CIFS service runs in an Active Directory forest and/or uses Kerberos to authenticate clients, this must be in a domain in the Active-Directory forest (configured in *Discovering the Active-Directory Forest (Kerberos)*, on page 3-9).

The CLI prompts for confirmation before creating the service; enter **yes** to proceed. (You can disable all such create-confirmation prompts with the terminal expert command.) This places you in gbl-cifs mode, from which you can share various namespace volumes.

For example, the following command sequence offers CIFS service at ac1.medarch.org:

```
bstnA(gbl)# cifs ac1.medarch.org
This will create a new CIFS service.

Create CIFS service 'ac1.medarch.org'? [yes/no] yes
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

From gbl-cifs mode, you must export at least one namespace volume and then enable the CIFS service, as described in the following subsections.

Sharing a Namespace Volume

If a namespace volume is configured for CIFS, you can offer it as a share through the CIFS-service configuration. You can offer as many CIFS shares as desired from a given namespace; each CIFS service supports one or more namespaces. The ARX can support a maximum of 16000 CIFS shares, total.

Use the export command to share a namespace volume through the current CIFS service:

```
export namespace vol-path [as share-name] [description description]
```

where

namespace (1-30 characters) can be any namespace that supports CIFS.

vol-path (1-1024 characters) is the path to one of the namespace’s volumes (for example, “/oneVol”) or volume sub paths (“oneVol/apps/myApps”).

as *share-name* (optional; 1-1024 characters) sets a share name for the volume. Potential CIFS clients see the ***share-name*** as an available share when, for example, they issue the **net view** command from a DOS prompt. If you omit this, the share name defaults to the volume name without the leading slash (/). By repeating this command with other share names, you can share the same volume under multiple names.

description ***description*** (optional; 1-64 characters) sets an optional description for the CIFS share. This describes the share in a Windows network browser (for example, **net view**). If the description includes any spaces, it must be surrounded by quotation marks (““).

Use show global-config namespace to see all available namespaces on the ARX. The CIFS service supports volumes from one or more CIFS namespaces.

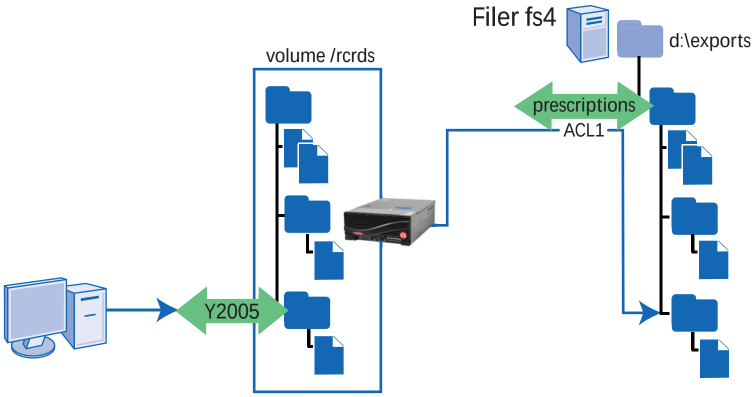
For example, the following command sequence adds a CIFS share to the CIFS service at ac1.medarch.org:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# show global-config namespace
;=============================== namespace ===============================
namespace medco
  protocol nfs3tcp
...
namespace medarcv
  protocol cifs
  cifs authentication kerberos
  cifs authentication ntlm
  cifs authentication ntlmv2
  ntlm-auth-server dc1
  ntlm-auth-server dc2
  proxy-user acoProxy2
  windows-mgmt-auth readOnly
  windows-mgmt-auth fullAccess
  windows-mgmt-auth snapViewers
  sam-reference fs2
  volume /lab_equipment
...
    enable
    exit

  volume /rcrds
    filer-subshares replicate
    modify
...
bstnA(gbl-cifs[ac1.medarch.org])# export medarcv /rcrds as ARCHIVES description “2-year-old
medical records”
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

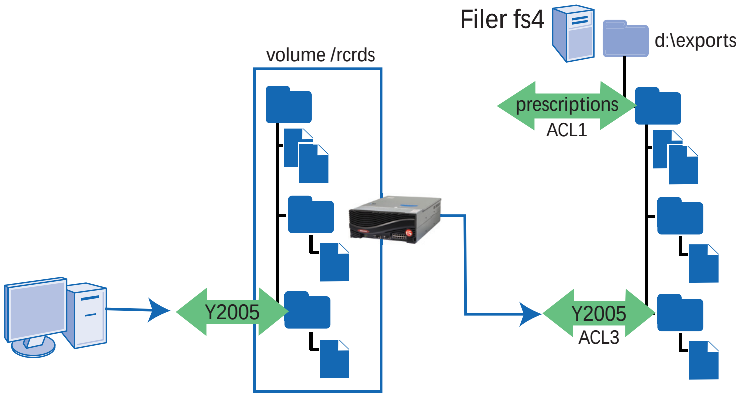
Exporting a Filer Subshare (and Using its ACL)

This section only applies to managed volumes. Skip all of the “subshare” sections if you are sharing a direct volume.

The CIFS service accesses each back-end share through its root, whether or not you export a directory below the root of the volume. For example, suppose you export “/rcrds/2005” as a *subshare* of the above “/rcrds” share. When a client accesses that subshare, the CIFS service goes through the top-level share at the back-end filer (for example, "\\fs4\prescriptions") and drops down to the /2005 directory. This bypasses any share-level ACL that the filer may have for its own version of the /2005 subshare.



You can configure the CIFS service to pass CIFS clients through to the same subshares, thus ensuring that they use the correct share-level ACL:



The volume and filers must be properly prepared before your CIFS service can offer this subshare service. A subshare must have the same ACL and position in the directory tree (relative to the share root) on every filer behind the volume. To continue the example, every filer share behind the "medarcv~/rcrds" volume must have

- a /2005 directory under its root,
- the same share name for it ("Y2005" in the above illustration) or a replicated share name created by the ARX volume (such as "_acopia_Y2005_9$"), and
- the same ACL ("ACL3," or its equivalent).

The managed-volume chapter discussed this preparation. Refer back to *Supporting Subshares and their ACLs*, on page 9-20, and *Replicating Subshares at all of the Volume's Filers*, on page 9-23.

With the volume properly prepared, you can use the filer-subshare flag in the export command:

```
export namespace vol-path as subshare-name filer-subshare
[description description]
```

where

namespace and

vol-path are both described above.

subshare-name must match the subshare name used at the back-end filers (“Y2005” in the above examples).

filer-subshare specifies that clients should pass through to the back-end subshare.

description *description* is also described above.

For example, the following command sequence adds the “Y2005” subshare to the CIFS service at ac1.medarch.org:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# export medarcv /rcrds/2005 as Y2005 filer-subshare
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Exposing Hidden Subshares

Some filer subshares can be hidden by having a dollar sign ($) at the ends of their share names (for example, “myshare$”). Most views of the filer’s CIFS shares do not show these names. The CIFS front-end service can expose a hidden subshare by using a slightly-different name for its front-end subshare; the back-end name without the “$” (such as “myshare”).

To expose hidden subshares, use the hidden keyword after filer-subshare:

```
export namespace vol-path as subshare-name filer-subshare hidden
[description description]
```

where

subshare-name is the name of the subshare in the CIFS service only (for example, “subshare1”). At the back-end filer(s), the same subshare name should already be configured with “$” at the end (“subshare1$”).

hidden is required.

For example, this command sequence adds the “old$” subshare and exports it under the exposed-share name, “old,” from the “ac1.medarch.old” service:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# export medarcv /rcrds/old as old filer-subshare hidden
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Ensuring the Consistency of All Subshares in a Managed Volume

This section applies only to CIFS subshares on managed volumes.

Each managed volume must have consistent subshares and subshare ACLs under all of its back-end shares. Consistency is required so that clients have the same access point and permissions no matter which back-end share contains their files and directories.

As of the current release, this situation is addressed automatically when subshare functionality is enabled using the filer-subshares command. The volume copies all subshare definitions from each share to the other shares. If necessary, the volume copies underlying directories as well. This explicit synchronization also occurs automatically any time that a new share is imported. As a result, it is unusual for subshares to become inconsistent.

In some cases, however, back-end subshares may be altered in some way by a user or application that modifies them independently of the ARX's control. (One possible example of this is anti-virus software, which may remove files or directories it determines to be hazardous.) In such cases, it may be necessary to synchronize the subshares with the front-end service in order to ensure that the configuration is consistent. In cases in which it is necessary, the volume replicates every subshare name, directory path (relative to the import-share root) and ACL. Replicated subshares use the same name as the source subshare, called the native subshare name, whenever possible.

The privileged-exec mode command sync subshares from-namespace replicates subshares and their corresponding ACLs between filers as needed, using the specified volume as the source of the subshare list that is to be synchronized.

The operation discovers all of the backend subshares behind a managed volume, synchronizes them (N-way across the relevant filers) so that all import shares have all subshares, and creates exports in the target service for that set of subshares.

The command's syntax is as follows:

```
sync subshares from-namespace ns volume vol to-service fqdn
[expose-hidden] [tentative]
```

If the expose-hidden option is used, consistent backend subshares for which the native name ends in "$" will be exported via the service with the "$" stripped away. If both a "$" and a non-"$" version of a backend subshare exist, the ARX will report this fact and use the "$" subshare only if the expose-hidden option is used.

If this replication process fails for any subshare, the front-end subshare is characterized as degraded, and clients cannot access of the files in the back-end subshare(s) that are not synchronized.

◆ **Note**

This command replaces the cifs export-subshares command that was available in earlier releases.

Displaying CIFS Service Transactions

The command show cifs-service transactions displays information about the CIFS services that are configured for a specified ARX. The following information is displayed for each CIFS service:

- source port
- version
- RPC
- duplicates

- status
- virtual volume/physical share

CIFS service transactions are the file access operations that take place over a CIFS service.

Adding New Subshares

This section only applies to managed volumes. Skip to the next section if you are sharing a direct volume.

The previous sections explained how to export pre-existing subshares, created on the back-end filers before their CIFS shares were imported. To add new subshares, you must directly connect to one of the back-end filers and create them there; a volume manages files and directories, but does not manage share definitions or ACLs.

◆ **Note**

This is contrary to standard best practices; direct access to the shares behind a managed volume can easily create inconsistencies in the volume's metadata. This exception is for new CIFS subshares only.

Stopping a CIFS Share

Use the no form of the export command to stop CIFS access to a namespace volume:

```
no export namespace volume [as share-name]
```

where

namespace (1-30 characters) is the namespace containing the CIFS share,

volume (1-1024 characters) is the name of a current CIFS share (for example, "alpha"), and

as *share-name* (optional, 1-1024 characters) stops access via the specified share name. If the volume is exported under more than one share name, the share is still supported under the remaining names.

For example, the following command sequence stops the CIFS server at ac1.medarch.org from sharing the "/cifstest" volume in the "medarcv" namespace:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# no export medarcv /cifstest
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Supporting MMC Browsing

As an alternative to managing the CIFS service from the CLI, Windows clients can use Windows-management applications to manage the service from a remote PC. Authorized clients can use the Microsoft Management Console (MMC) and similar applications to export volumes, list open files, close client sessions, and so on.

In MMC's interface for adding new shares, MMC offers an option to "Browse" a CIFS filer's storage and share the filer's volumes. The CIFS service does not support this type of browsing by default; from gbl-cifs mode, use the browsing command to enable it. MMC users can then use the MMC-browsing mechanism to select volumes behind the CIFS service and possibly share them:

```
browsing namespace
```

> where ***namespace*** (1-30 characters) is a namespace that clients can access through Windows-management applications like MMC. A CIFS service can export volumes from multiple namespaces; if you have used the export command already, this must be one of the same namespaces that you specified previously.

Only authorized Windows clients can manage the CIFS service at all. You can configure a group of CIFS clients with Windows-management access (recall *Authorizing Windows-Management (MMC/Snapshot) Access*, on page 3-25) and associate it with this service's namespace (also recall *Opening Windows-Management Access (optional, MMC/Snapshots)*, on page 7-16). These clients can use MMC and other remote-management applications to fully manage the namespace. If they are authorized to modify shares, they can use the MMC-browsing feature you configure here.

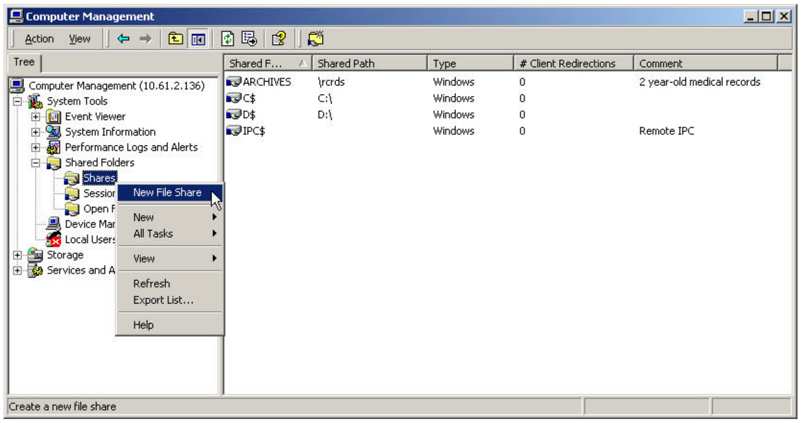
For example, the following command sequence allows authorized clients of the "ac1.medarch.org" service to use the MMC-browsing feature:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# browsing medarcv
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Client Experience: Using MMC to Share a Volume

A properly-enabled MMC client can share out a volume from this CIFS service. For example, the following client session adds a share to the "ac1.medarch.org" service from a Windows 2000 machine. The session starts from Start -> Control Panel -> Administrative Tools -> Computer Management, where you connect to the VIP for the service. This shows a virtual network drive, "C$," that serves as a container for all of the managed volumes in the namespace. It also shows a separate drive for each direct volume, "D$," "E$," and so on. Any of the CIFS service's shares (exported through the CLI or GUI) also appear here.

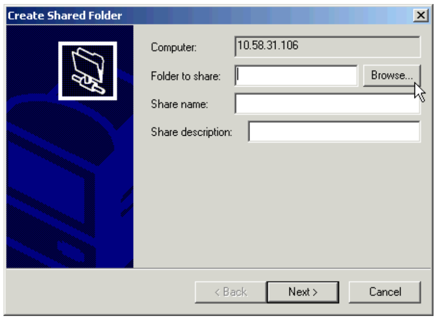
From the MMC snap-in interface, you select System Tools -> Shared Folders -> Shares, then pull down a menu option to add a new share:



Shared F...	Shared Path	Type	# Client Redirections	Comment
ARCHIVES	\rcrds	Windows	0	2 year-old medical records
C$	C:\	Windows	0	
D$	D:\	Windows	0	
IPC$		Windows	0	Remote IPC

Create a new file share

A pop-up appears. You opt to browse the CIFS service for folders to share:



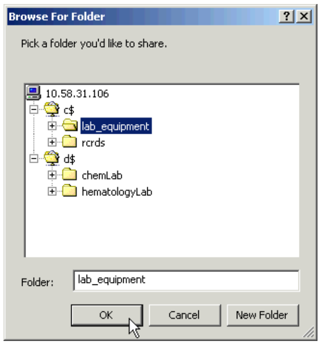
Computer: 10.58.31.106

Folder to share:

Share name:

Share description:

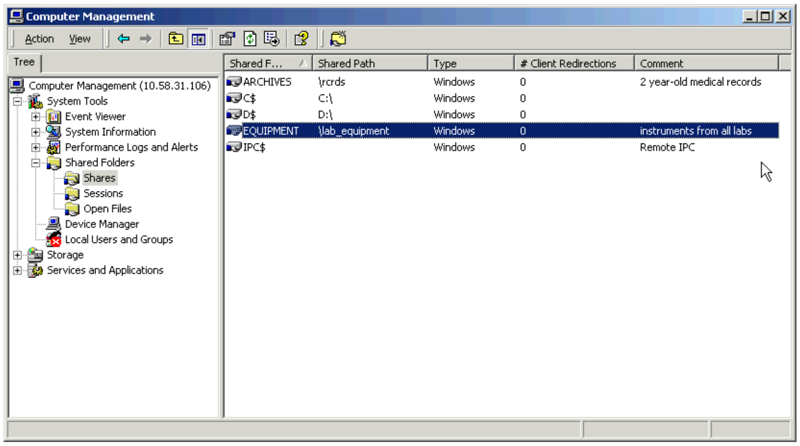
This shows all managed volumes in the CIFS service's namespace under the C drive. This listing is the feature that you enable with the browsing command. Each direct volume in the namespace appears as another drive. In this example, the two managed volumes appear as folders under the C drive, one direct volume appears as the D drive:



Pick a folder you'd like to share.

Folder: lab_equipment

You use the interface to export the other managed volume with the share name, “EQUIPMENT.” The result is visible from MMC:



Shared F...	Shared Path	Type	# Client Redirections	Comment
ARCHIVES	\rcrds	Windows	0	2 year-old medical records
C$	C:\	Windows	0	
D$	D:\	Windows	0	
EQUIPMENT	\lab_equipment	Windows	0	instruments from all labs
IPC$		Windows	0	Remote IPC

Disallowing MMC Browsing

Use no browsing to disable MMC browsing for the current CIFS service:

```
no browsing
```

For example, the following command sequence prevents MMC browsing in the CIFS service, “beta_service:”

```
bstnA(gbl)# cifs beta_service
bstnA(gbl-cifs[beta_service])# no browsing
bstnA(gbl-cifs[beta_service])# ...
```

Setting a Server Description (optional)

You can optionally set the CIFS-service description that will appear in Windows network browsers. The description appears in the “Remarks” column when you issue a Windows net view command:

```
U:\>net view
Server Name            Remark
-------------------------------------------------------------------------------
\\MEDSERVER
\\ARCHIVE1         A full tree of 5-year-old records
```

Use the description command to set a description for this CIFS service:

```
description description
```

where ***description*** (1-48 characters) is a quoted string to describe this CIFS service.

For example, the following command sequence sets the CIFS-service description for the global server at “ac1.medarch.org:”

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# description “medical histories and records”
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Removing the Description

Use the no form of the description command to remove the description for this CIFS service:

```
no description
```

For example, the following command sequence removes the default description for the global server at “ac1.medarch.org:”

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# no description
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Enabling SMB Signing with Clients (Optional)

SMB signing is the process of adding a digital signature to the header of each server message block (SMB) that passes between a CIFS server and its clients. This protects against “man-in-the-middle” attacks, but increases the processing overhead for each session and therefore reduces the overall performance of the ARX.

To control SMB signing between the ARX and its back-end filers, use the cifs filer-signatures command in gbl-ns mode. This is described earlier, in *Using SMB Signing with Back-End Filers (optional)*, on page 7-18.

Use the signatures command to enable SMB signing for connections between this CIFS service and its clients:

```
signatures [required]
```

> where **required** (optional) makes the CIFS service refuse any client connection that does not support SMB signing. If you omit this option, the service always attempts to connect with SMB signing enabled, but connects to a CIFS client even if the client machine does not support SMB signing.

For example, the following command sequence shows a preference (but not a requirement) for SMB signing between the “ac1.medarch.org” service and its clients:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# signatures
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Disabling SMB Signing with Clients

Use the no form of the signatures command to disable SMB signing between this CIFS service and its clients:

```
no signatures
```

For example, the following command sequence stops SMB signing for “labtests.medarch.org:”

```
bstnA(gbl)# cifs labtests.medarch.org
bstnA(gbl-cifs[labtests.medarch.org])# no signatures
bstnA(gbl-cifs[labtests.medarch.org])# ...
```

Setting Character Encoding For WINS

When a volume from a CIFS namespace is exported through a virtual server, the virtual server may register its NetBIOS name with a WINS server. Use the wins-name-encoding command to set the character encoding expected by the local WINS server:

```
wins-name-encoding {iso–8859–1 | utf–8 | shift-jis | cp932 | euc-jp | ksc5601}
```

where

iso–8859–1 is ISO 8859–1 (Latin1, single-byte) character encoding,

utf–8 specifies UTF–8 (Unicode) character encoding,

shift-jis (Microsoft Shift-JIS) supports Japanese characters,

cp932 is Code Page 932, or Windows-31J (Japanese) character encoding. This is the Microsoft version of Shift_JIS.

euc-jp is variable-width Japanese character encoding, based on the JIS X 0208, JIS X 0212, and JIS X 0201 standards.

ksc5601 supports Korean characters.

The default is ISO 8859–1 (Latin1).

For example, this command sequence sets character encoding to “Shift JIS” (Japanese) for CIFS NETBIOS names for the ‘insur’ multi-protocol namespace:

```
bstnA(gbl)# namespace insur
bstnA(gbl-ns[insur])# wins-name-encoding cifs shift-jis
bstnA(gbl-ns[insur])# ...
```

Controlling Client Access to Offline Shares

When a Windows client uses *offline access* for a file on a remote CIFS share, Windows creates a local copy of the file for the client. The client can use that local copy whenever the client machine is disconnected from the CIFS share, and can later *sync* the local copy with the original whenever the CIFS-share connection is up.

By default, clients can select directories and files for offline access manually. You can use the export offline-access command to also automatically enable offline access for any file the client opens (with or without network optimization), or to disable all offline access.

The command syntax is:

```
export offline-access share-name {manual|auto|auto-local|none}
```

Where:

- ***share-name*** (1-1024 characters) is the name of a share or subshare from this CIFS service, established with the *export (gbl-cifs)* command. You can use *show cifs-service* *fqdn* for a list of all shares and subshares in the *fqdn* service.

- **manual | auto | auto-local | none** is a required choice. This maps to the offline settings in the MMC interface for managing CIFS shares. Use this command in place of those options:
    - **manual** means that clients must set the option for offline files and directories themselves. This maps to the MMC option (in Windows 7), “Only the files and programs that users specify are offline.”
    - **auto** indicates that client software is allowed to automatically enable offline access for every file the client opens. Client software is also allowed to automatically sync its local files with the files on the CIFS share whenever the connection is up. This can produce a great deal of network traffic between clients, the ARX, and the back-end filers behind this CIFS share. It maps to the Windows 7 MMC option, “All files and programs that users open from the shared folder are automatically available offline.”
    - **auto-local** extends the **auto** feature. This maps to the same MMC option above together with an additional “Optimized for performance” option. This permits the client software to reduce its network usage by downloading all executable files immediately after connecting. For newer Windows clients, this is equivalent to the **auto** feature.
    - **none** means that the client software is not permitted to offer any offline-access options to the client. This maps to the MMC option, “No files or programs from the shared folder are available offline.”

Use the no form of the command to revert to the default setting:

```
no export offline-access share-name
```

Joining a CIFS Service to an Active Directory Domain

A front-end CIFS service must be registered with an Active Directory domain to acquire the necessary authentication and delegation configuration. This is done using the domain-join CLI command in gbl-cifs mode.

This section describes the considerations related to joining a domain, as well as the use and behavior of the domain-join command.

Authentication Protocols

The ARX supports authentication both for front-end services (CIFS, in this case) and for back-end filers. Back-end authentication refers to the ARX authenticating to a back-end filer, either on behalf of a client that has authenticated to a front-end CIFS service or on behalf of one of its own administrative services, such as back-end file monitoring or a policy. Authentication on behalf of one of its own services employs a proxy user that is not related to any client connection.

Each client negotiates the authentication protocol with the CIFS service when it initiates a CIFS session. If all protocols are configured and functional, the CIFS service will indicate its *preference* for Kerberos, but will accept NTLM or NTLMv2 from clients if that is what they're using.

To enable Kerberos authentication by the CIFS service, you must join the CIFS service to the Active-Directory (AD) domain using the domain-join command. This process is similar to adding client computers to the AD domain: the action causes the DC to declare the CIFS service as trusted-for-delegation (refer to *Constrained Delegation Support*, on page 11-28). The CIFS service uses this authority to access back-end filers on behalf of its clients.

◆ **Note**

Trusting an ARX service for delegation poses no security threat to your network. Kerberos authentication was designed with delegation in mind to provide a clean way of carrying identity through n-tiered application systems. For more information, refer to IETF RFC 1510 or the Microsoft white paper on Kerberos authentication (http://www.microsoft.com/windows2000/techinfo/howitworks/security/kerberos.asp).

The domain-join operation does not preclude any clients from authenticating with NTLM or NTLMv2; the CIFS service can support both authentication protocols concurrently. You can use this feature for a network transition from NTLM to Kerberos.

For Windows networks that use Kerberos authentication, the CIFS service can provide proxy authentication for its clients. You can use Kerberos only if the namespace behind the CIFS service is configured for it; see *Using Kerberos for Proxy-User Authentication*, on page 7-16. Additionally, the service's FQDN must belong to a domain in the pre-configured Active-Directory forest (refer to *Discovering the Active-Directory Forest (Kerberos)*, on page 3-9).

◆ **Note**

If the domain has been joined and constrained delegation is in use, then regardless of which protocol clients use to authenticate to the front-end service, the ARX always authenticates to the back-end with Kerberos. Otherwise, the ARX will authenticate to the back-end using the same protocol that the clients used to authenticate on the front-end.

Synchronizing CIFS Delegation Settings

Use the sync cifs delegation command in privileged-exec mode to force the synchronization of an ARX CIFS service with its delegation settings in Active Directory. This is useful if you know that the delegation settings for one or more CIFS services have been modified at the domain controller, and you don't want to wait for up to ten minutes for the ARX to synchronize with the domain controller automatically.

The command's syntax is:

```
sync cifs delegation {fqdn | all}
```

where:

fqdn specifies the fully-qualified domain name for the CIFS service for which you want to synchronize the delegation settings, and

all indicates that you want to synchronize the delegation settings for all of the ARX's existing CIFS services.

For example:

```
bstnA# sync cifs delegation bgh.MEDARCH.ORG
```

synchronizes the delegation settings for the CIFS service at bgh.MEDARCH.ORG.

Constrained Delegation Support

Delegation is a means by which a server is configured to pass along a client's identity and credentials to a second server, accessing a resource or service at that second server on behalf of the client. The ARX software supports constrained delegation, in which identity and credentials are passed along for explicitly specified servers or services only.

◆ **Note**

The use of constrained delegation is strongly recommended.

The ARX supports these constrained delegation use cases:

- CIFS client authentication to a CIFS service using a Kerberos ticket.
- CIFS client authentication to a CIFS service using NTLM or NTLMv2.

In each case, the CIFS service uses the computer account's credentials to obtain a service ticket to authenticate to filers on behalf of the front end user, thereby preserving the practice of carrying the identity of the front end user to the back end filers that are virtualized.

Protocol transition is employed when constrained delegation is used with NTLM or NTLMv2 on the front-end, where the "transition" that takes place is the transference of the identity and credentials from NTLM/NTLMv2 to Kerberos.

Using the domain-join Command

Join a CIFS service to a domain using the gbl-cifs domain-join command as shown here:

```
domain-join domain-name [ou "organizational-unit"] [delegation
{constrained | unconstrained | none}] [timeout seconds]
```

where

domain-name (1-256 characters) identifies the DC domain. This domain must be defined in the AD forest; see *Discovering the Active-Directory Forest (Kerberos)*, on page 3-9.

organizational-unit (optional, 1-512 characters) is the organizational unit (OU) to join. An *OU* is a group of similar accounts or machines that is managed by a particular administrator. The default is "Computers" at the root of the domain. Quote the OU; OUs often contain spaces. For a nested OU, use a backslash (\) to separate each OU layer (for example, "Virtual Servers\lab"). If the OU does not exist, the domain-join operation fails.

Optionally, you can specify a timeout value for the domain-join operation, expressed in seconds, in case the DC is too slow to respond before the command times out.

Minimum Windows Privileges Required For domain-join

After you issue the domain-join command, the CLI prompts for a username and password. These are the credentials to be presented to the DC, called the *domain-joiner* credentials. In some sites, the domain-joiner credentials are sufficiently privileged to create a machine account on the DC and raise all of the proper flags. In other sites, a more-privileged user creates the machine account before the domain-joiner runs this command. These next subsections describe the minimal access privileges required to join the domain with either a new machine account or a pre-created account.

New Machine Account

If the domain-joiner has the following privileges, the domain-join operation can create a new machine account for the CIFS service. A pre-created machine account is unnecessary.

- In the "Computers" OU for the domain, the user must have permission to "Create Computer Objects."

  On a Windows Server 2003 DC, you can set this in the "Active Directory Users and Computers" interface.

- The user must be able to "Enable computer and user accounts to be trusted for delegation."

  On a Windows Server 2003 DC, you can find this under "Domain Controller Security Policy."

This is only required if you use some form of delegation, either constrained or unconstrained. It is unnecessary if you specify delegation none.

Pre-Created Machine Account

If someone in another group pre-creates the machine account at the DC, that person must have the minimal privileges above. On a Windows Server 2003 DC, that person creates a new machine account in the "Active Directory Users and Computers" interface. The machine account requires the following minimal settings:

- Set the machine account's "Delegation" tab as desired. We recommend "Trust this computer for delegation to specified services only" together with "Use any authentication protocol." This is what the domain-join operation sets when it creates the account with the delegation constrained options.
- Add the domain-joiner's user account into the machine account's "Security" tab.

  Give the following permissions to that user:

  - Read
  - Write
  - Reset Password

Do not use any of the domain-join command's delegation options with a pre-created machine account. The delegation is already set at the DC, as described above.

Joining a Domain When Using Constrained Delegation

Join the domain with the use of constrained delegation using the gbl-cifs mode command:

domain-join *domainname* [ou *ouname*] delegation constrained

If the account does not exist, this command sets the computer account's delegation property to trusted-for-constrained delegation.

Joining a Domain Without Specifying a Delegation Option

Join the domain without specifying a delegation option using the gbl-cifs mode command:

domain-join *domainname* [ou *ouname*]

The purpose of this command is for the ARX administrator to be able to create an account by using an AD account that does not have the necessary privilege to set the account trust property. The trust property can be set by an AD administrator at a later time.

The CIFS service will not accept Kerberos authentication with this setting.

If the computer account is pre-created by an Active Directory administrator, the domain-join does not change the delegation property of the account. If the account does not exist, the domain-join command creates it and sets the account delegation property to trust-for-unconstrained-delegation.

Switching a Joined Service to Constrained Delegation

If the CIFS service is already joined to the domain and you want to upgrade to constrained delegation, you can upgrade at the DC without disrupting your CIFS clients. At the DC, use the "Active Directory Users and Computers" interface to access the machine account for *fqdn*. (The machine account name is the first part of the *fqdn*: for example, the machine account name for "ac1.medarch.org" is "ac1".) In the machine account's properties, access the machine account's "Delegation" tab and enter

- “Trust this computer for delegation to specified services only,”
- “Use any authentication protocol,” and
- the “delegate to” list at the bottom, which requires all the filers behind this CIFS service. See below if you are not aware of all the filers behind this service.

The CIFS service probes the DCs every few minutes to find these settings, so it may not be up-to-date right after you change them. To verify success, wait several minutes and then use the probe delegate-to *fqdn* command; this probes the DC and confirms that all the necessary filers are on the “delegate to” list. If any of these filers show up as “failed” on that list, you can go back to the DC and enter those filers there.

Once this is done for all of your CIFS services, you can remove all ARX Secure Agent applications from your DCs. When constrained delegation is active in all CIFS services, the Secure Agent applications are ignored. For more information about Secure Agent, see the *ARX® Secure Agent Installation Guide*.

Disabling Delegation

Disable delegation using the gbl-cifs mode command:

```
domain-join domainname [ou ouname] delegation none
```

Example of the domain-join Command In Use

For example, the following command sequence joins the domain “MEDARCH.ORG” with constrained delegation under the username “acoadmin:”

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# domain-join MEDARCH.ORG delegation
constrained
Username: acoadmin
Password: ******

% INFO: Service 'ac1' successfully joined the domain using a
pre-created computer account.


% INFO: Service 'ac1' joined the domain with delegation type set to
Constrained. Only certain selected services are allowed to be
delegated on 'bgh'. The Active Directory administrator must configure
the services 'bgh' may delegate to, which are the filer(s) this CIFS
service virtualizes.

bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Using an SPN With Constrained Delegation in a Two-Tier Configuration

There are some special considerations for using an SPN in a two-tier configuration with delegation when a client is connected to a second-tier server that uses constrained delegation to connect to the ARX VIP. In a two-tier configuration, when you select a CIFS service, you add two CIFS SPNs for the VIPs: one with CIFS in all uppercase and one with CIFS in all lowercase. There is an issue related to how the SPN is set up during a domain join. During a domain join, four SPNs are set up for each computer account:

- CIFS/shortname
- CIFS/FQDN
- HOST/shortname
- HOST/FQDN

The default lowercase SPN (cifs/shortname) is set by Microsoft when a computer account is joined to a domain, and this SPN does not show up in the setspn -l list. On the delegation list page, when selecting CIFS account, both cifs/shortname and CIFS/shortname are selected. Since both are the same (case insensitive), there is a conflict.

To resolve this conflict, on the DC Active Directory Users and Computers page, select computer name, then property > delegation > Add Delegation List. On this Delegation page you must click to expand the list and remove one of these two names to make it work. Please note, the known supported servers include MSFT IIS and SFTP server.

Enabling CIFS Service

The next step in CIFS configuration is to enable it. Use the enable command from gbl-cifs mode to activate the CIFS service:

```
enable
```

For example, the following command sequence enables CIFS for the global server at “ac1.medarch.org:”

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# enable
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Disabling CIFS

Disabling CIFS shuts down all CIFS connections to the global server. From gbl-cifs mode, use no enable to disable CIFS.

```
no enable
```

For example, the following command sequence disables CIFS for the “ac1.medarch.org” global server:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# no enable
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Listing the Filers Behind a CIFS Service

After a CIFS service joins its domain using constrained delegation (via the *domain-join* command), you must inform the DC of the list of filers to which the CIFS service can delegate. Use the probe delegate-to command to display a list of all the filers behind a specified CIFS service, along with their current delegation status.

◆ **Note**

The probe delegate-to command is relevant only for a CIFS service with constrained delegation.

The command's output displays the exact username and domain used for the probe, followed by a table of probe results. The table contains one row per filer behind the CIFS service, with each row containing the following fields:

- External-Filer is the name of the back-end filer. You can use the *show external-filer* command to list all external filers on the ARX.
- SPN is the service-principal name (SPN) of the back-end filer. This is the SPN that the ARX discovered, or the one that was set with the *spn* command.
- Probe-SPN is the service-principal name (SPN) of the back-end filer. This is the SPN that was entered at the DC.
- Delegation Status is Ok if the DC allows the front-end CIFS service to delegate to this filer. A Failed status indicates that you need to take further action. Here are examples of common failed-status messages:
  - Failed-Not in delegate to list means that the front-end CIFS service cannot delegate to this filer. Go to one of the DCs for the service's domain and add this filer to the Delegate list for the CIFS service. Configure the CIFS service to delegate "cifs" to the filer.
  - Failed - Filer spn is not configured nor discovered indicates that the filer's service principal name (SPN) is unknown. You can use *show exports* to automatically discover a filer's SPN, or *spn* to manually set it.
  - Failed-Filer not in the same domain means that the back-end filer is not joined to the same domain as the CIFS service. The CIFS service belongs to the Windows domain of its backing namespace: use *show cifs-service* *cifs-service* to find this Windows domain for a given *cifs-service*. The filer must be joined to the same Windows Domain for constrained delegation to function.

Examples of Listing Filers Behind CIFS Services

This section provides two examples of using the probe delegate-to command.

The example that follows shows the probe delegate-to command being executed relative to a CIFS service named "insur.medarch.org", a user account named "juser", and a domain named "MEDARCH.ORG":

Figure 11.1 *Sample Output: probe delegate-to insur.medarch.org*

```
bstnA# probe delegate-to insur.medarch.org user juser domain MEDARCH.ORG

User used for probe:  juser@MEDARCH.ORG

  External Filer : nas1
    Filer SPN    : ntap820$@MEDARCH.ORG
    Probe SPN-1  : CIFS/ntap820@MEDARCH.ORG -- OK
    Probe SPN-2  : CIFS/ntap820.MEDARCH.ORG@MEDARCH.ORG -- OK

  External Filer : nasE1
    Filer SPN    : engdm$@MEDARCH.ORG
    Probe SPN-1  : CIFS/engdm@MEDARCH.ORG -- OK
    Probe SPN-2  : CIFS/engdm.MEDARCH.ORG@MEDARCH.ORG -- OK
```

Examples of Listing Filers Behind CIFS Services

The example that follows shows the probe delegate-to command being executed relative to a CIFS service named “swimed.atlantic.me.org”:

Figure 11.2 *Sample Output: probe delegate-to swimed.atlantic.me.org*

```
bstnA# probe delegate-to swimed.atlantic.me.org

User used for probe:  administrator@ATLANTIC.ME.ORG

  External Filer : smb-1
    Filer SPN    : VM-SWP2008S-13@ATLANTIC.ME.ORG
    Probe SPN-1  : CIFS/VM-SWP2008S-13@ATLANTIC.ME.ORG -- OK
    Probe SPN-2  : CIFS/VM-SWP2008S-13.atlantic.me.org@ATLANTIC.ME.ORG -- OK
```

Using Dynamic DNS (Kerberos)

Every front-end CIFS service that uses Kerberos must be registered with the network’s Domain Name Service (DNS), which maps IP addresses to FQDNs. In each AD domain, one or more *name servers* take these DNS registrations; often, DNS runs on the DC itself. You can manually update the local name server with the hostname-to-IP mapping for the CIFS service, or you can configure the service to use *dynamic DNS*. With dynamic DNS, an enabled CIFS service automatically registers its hostname-to-IP mapping, and updates it whenever a failover or configuration change makes an update necessary.

RFCs 1034 and 1035 define basic DNS, and RFC 3645 defines Microsoft-specific authentication extensions to for dynamic DNS. The ARX implementation of dynamic DNS adheres to all of these RFCs.

Before you use dynamic DNS, the name server(s) for this service’s Windows Domain must be included in the AD forest. For instructions on identifying the AD forest’s name servers, recall *Identifying a Dynamic-DNS Server*, on page 3-10. The Windows Domain for this CIFS service is determined by the service’s global server; use show global server to see the domain (recall *Showing One Global-Server*, on page 10-13).

Use the dynamic-dns command to choose a name for this service and register it with a name server. If the CIFS service is enabled, this sends an “A” (Address) record that maps the virtual server’s VIP to a host name:

```
dynamic-dns host-name
```

> where ***host-name*** (1-255 characters) is a host name you choose for this CIFS service. This is mapped to the virtual server’s VIP in the “A” record. If you enter a host name without any domain (for example, “myservice”), the global server’s Windows domain is appended to it. If you specify a domain (for example, “myservice.myco.net”), the domain must match the Windows domain for the global server.

This command has no effect if the CIFS service is disabled.

You can repeat the command to enter additional aliases for this CIFS service. All of these host names map to the CIFS service’s VIP; Windows clients can use any of them to access the CIFS service.

For example, this command sequence registers two host names, “test” and “ac1,” for the current CIFS service:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# dynamic-dns test
bstnA(gbl-cifs[ac1.medarch.org])# dynamic-dns ac1
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Replacing Back-End Services

Through DNS aliasing, you can transparently replace the CIFS services on existing back-end filers. For example, suppose you had the following CIFS filers and shares before the ARX was installed:

- \\fs1.medarch.org\chem_results
- \\fs2.medarch.org\bulkstorage
- \\fs5.medarch.org\xraysScanners

You can import each of these shares into a single namespace, where each share is in a separate volume. From gbl-cifs mode, you can then export each CIFS volume under its original share name (“chem_results,” “bulkstorage,” and “xraysScanners,” respectively). Finally, you can use the dynamic-dns command to register all three of the original host names as DNS aliases for the CIFS service (“fs1,” “fs2,” and “fs5”). Clients can then access all three shares using the same host names and share names. For example:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# dynamic-dns fs1
bstnA(gbl-cifs[ac1.medarch.org])# dynamic-dns fs2
bstnA(gbl-cifs[ac1.medarch.org])# dynamic-dns fs5
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

◆ **Note**

To support Kerberos authentication at these DNS aliases, you must also register them as service principal names (SPNs) for the CIFS service. This was described in an earlier chapter; recall **Setting SPNs (CIFS/Kerberos Only)***, on page 10-9.*

No further action is required for the CIFS-service FQDN itself. Windows clients can connect to the service's FQDN and authenticate with Kerberos without any additional configuration.

Update Schedule for DNS "A" Records

The CIFS service updates the DNS database as soon as you enter the dynamic-dns command. It re-registers the "A" record once per day at 1:00 AM (local time), whenever the ARX reboots or fails over, whenever the VIP for this service changes, or if ever the VIP or CIFS service is deleted. The CIFS service only performs DNS registrations when it is enabled.

To update all DNS records immediately, you can exit to priv-exec mode and use the dynamic-dns update command:

```
dynamic-dns update [fqdn]
```

where ***fqdn*** (optional: 1-128 characters) narrows the scope of the update to one CIFS service. This is the FQDN that identifies the CIFS service and its global server. If you omit this, the ARX updates all host names for all enabled CIFS services; this can be network-intensive.

The CLI prompts for confirmation if you choose to update for all CIFS services. Enter **yes** to continue.

For example, the following commands exit to priv-exec mode and update all of the DNS entries for a particular CIFS service, "ac1.medarch.org:"

```
bstnA(gbl-cifs[ac1.medarch.org])# end
bstnA# dynamic-dns update ac1.medarch.org
bstnA# ...
```

Removing a Host Name

You can use the no dynamic-dns command to remove one host name from the current CIFS service. This causes the CIFS service to withdraw all references to this host name from DNS. If you remove all registered host names for this CIFS service, you must manually update the domain's DNS server to support Kerberos.

```
no dynamic-dns host-name
```

where ***host-name*** (1-255 characters) is the host name to remove.

As above, this command only functions while the CIFS service is enabled.

For example, this command sequence removes "test" as a DNS alias for the current CIFS service:

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# no dynamic-dns test
bstnA(gbl-cifs[ac1.medarch.org])# ...
```

Showing Dynamic-DNS Status

The show dynamic-dns command displays the status of all host-name registrations:

```
show dynamic-dns [fqdn]
```

where ***fqdn*** (optional: 1-128 characters) is the fully-qualified domain name (for example, www.company.com) for the CIFS service's global server. If you omit this, the output includes all CIFS services.

You can invoke this command from any mode.

For each CIFS service, this lists all host names, whether or not their "A" records were properly registered, the time of the last registration, and the IP address of the DNS server that accepted the record. For example, this lists the five host names associated with the "ac1.medarch.org" service, using the single VIP, 192.168.25.15, and registered successfully at the local DNS server:

```
bstnA> show dynamic-dns ac1.medarch.org

Svc     Global Server                  Domain Name
-----------------------------------------------------------------------------
CIFS    ac1.MEDARCH.ORG                MEDARCH.ORG

  Status   Host Name                                         VIP
           Operation      Retries   Last Update              DNS Server
  ---------------------------------------------------------------------------
  OK       ac1.MEDARCH.ORG                                   192.168.25.15
           Add                0     Tue Jan 25 02:03:01 2011  192.168.25.102

  OK       fs1.MEDARCH.ORG                                   192.168.25.15
           Add                0     Tue Jan 25 02:03:01 2011  192.168.25.102

  OK       fs2.MEDARCH.ORG                                   192.168.25.15
           Add                0     Tue Jan 25 02:03:01 2011  192.168.25.102

  OK       fs5.MEDARCH.ORG                                   192.168.25.15
           Add                0     Tue Jan 25 02:03:02 2011  192.168.25.102

  OK       insur.MEDARCH.ORG                                 192.168.25.15
           Add                0     Tue Jan 25 02:03:02 2011  192.168.25.102

bstnA>
```

Clearing Records of Failed Deletes

The CIFS service performs two dynamic-DNS operations: *add* and *remove*. The dynamic-dns command triggers an add operation, and a remove occurs after no dynamic-dns. If an add operation fails, the service retries once per minute indefinitely. If a remove operation fails, the service stops retrying after the 15th attempt. After the 15th failure, the remove operation gets a "Failed" status in the show dynamic-dns output; you will have to remove the "A" record at the name server itself. For your convenience, these records remain in the show dynamic-dns output until someone clears them.

From priv-exec mode, you can use the clear dynamic-dns command to clear all failed deletes from the show output:

```
clear dynamic-dns
```

The CLI prompts for confirmation before deleting all the records on the system; enter **yes** to proceed.

For example, the following command sequence shows a failed remove operation (highlighted in bold text), clears it, and shows that it is cleared:

```
bstnA# show dynamic-dns

Svc     Global Server                   Domain Name
-----------------------------------------------------------------------------
CIFS    ac1.MEDARCH.ORG                 MEDARCH.ORG

  Status  Host Name                                           VIP
          Operation      Retries   Last Update                DNS Server
  ---------------------------------------------------------------------------
  Failed  test.MEDARCH.ORG                                    192.168.25.15
          Remove            15     Wed Oct  4 07:09:33 2006   192.168.25.102

  OK      ac1.MEDARCH.ORG                                     192.168.25.15
          Add                0     Wed Oct  4 06:56:24 2006   192.168.25.104

  Retry   fs7.MEDARCH.ORG                                     192.168.25.15
          Add               19     Wed Oct  4 07:12:23 2006   192.168.25.104

bstnA# clear dynamic-dns
Clear failed dynamic DNS entries? [yes/no] yes
bstnA# show dynamic-dns

Svc     Global Server                   Domain Name
-----------------------------------------------------------------------------
CIFS    ac1.MEDARCH.ORG                 MEDARCH.ORG

  Status  Host Name                                           VIP
          Operation      Retries   Last Update                DNS Server
  ---------------------------------------------------------------------------
  OK      ac1.MEDARCH.ORG                                     192.168.25.15
          Add                0     Wed Oct  4 06:56:24 2006   192.168.25.104

  Retry   fs7.MEDARCH.ORG                                     192.168.25.15
          Add               19     Wed Oct  4 07:12:23 2006   192.168.25.104

bstnA# ...
```

Changing the ARX's Machine-Account Keys (Kerberos)

The local AD policy may require machine accounts (computers joined to the domain, such as this CIFS service) to periodically change their secret keys. The secret key, or password, is used to authenticate the CIFS service with the DC. If a CIFS service's key expires, it can no longer authenticate any clients with Kerberos. By default, DCs do not assign an expiration period to their machine-account keys; this is an optional AD-policy setting.

For a site where the machine-account keys have a maximum age, you can use the cifs rekey all command to reset all of the ARX's machine-account keys before they expire.

◆ **Note**

All CIFS services stop and restart as they change their keys. This causes an outage of several seconds. This outage is brief enough that it should not be perceptible to the volume's clients.

You enter the cifs rekey all command from priv-exec mode. If this switch has a redundant peer, enter this command on the active switch:

```
cifs rekey all
```

The CLI prompts you with a warning about the service outage; enter **yes** to proceed. Then the CLI lists the CIFS services that are affected by the change.

For example, the following command sequence resets all machine-account keys for the ARX:

```
bstnA(gbl)# end
bstnA# cifs rekey all

WARNING: Confirming this operation will cause the ARX to reset the
machine account password for all Kerberos-enabled CIFS services on the
switch. This will cause an outage of 30-60 seconds for all CIFS/NFS
services on the switch.

Proceed [yes/no] yes
  The machine account password for CIFS service 'ac1.medarch.org'
successfully changed.
bstnA# ...
```

Scheduling Regular Rekey Events

A Windows DC sets a maximum age for its machine-account keys, so the keys expire on a regular basis. You can use the at command to regularly refresh all of your CIFS-service keys before they expire. The at command is in cfg mode.

For example, the following command sequence runs the cifs rekey all command every 89 days. This would be an effective schedule for a site where machine-account keys expire every 90 days:

```
bstnA(gbl)# end
bstnA# config
bstnA(cfg)# at 03:30:00 every 89 days do "cifs rekey all"

  The scheduled execution time for CLI command is: 9/8/08 3:30 AM.

bstnA(cfg)# ...
```

Recovering from a Password Change

If a CIFS service's password expires before you use the cifs rekey command, you must remove the service configuration, rebuild it, and rejoin it to the AD domain. The *ARX CLI Maintenance Guide* contains detailed instructions for leaving and rejoining an AD domain.

Listing CIFS Services

Use the show cifs-service command to list all global servers that offer CIFS services:

```
show cifs-service
```

For example:

```
bstnA> show cifs-service

Service Name                Description
---------------------------------------------------
ac1.MEDARCH.ORG             insurance-claim records
labtests.MEDARCH.ORG        tests and records for lab animals

bstnA>
```

Focusing on One CIFS Service

To focus on a particular CIFS front-end service, specify the FQDN for the service in the show cifs-service command:

show cifs-service *fqdn*

where ***fqdn*** (1-128 characters) is the fully-qualified domain name (for example, www.company.com) for the global server.

This shows a configuration summary followed by a table of CIFS shares.

For example:

```
bstnA> show cifs-service ac1.medarch.org

Service Name:          ac1.MEDARCH.ORG
Domain Join:           Joined to MEDARCH.ORG
Account Name:          ac1$
Delegation:            Constrained, Any Protocol
Delegate To:           cifs/PV770N
                cifs/PV770N.MEDARCH.ORG
                cifs/VM-PV770N-01
                cifs/VM-PV770N-01.MEDARCH.ORG
                cifs/VM-SWP2003S1-5
                cifs/VM-SWP2003S1-5.MEDARCH.ORG
                cifs/VM-SWP2003S2-04
                cifs/VM-SWP2003s2-04.MEDARCH.ORG
                cifs/VM-SWP2008S-01
                cifs/VM-SWP2008S-02
                cifs/VM-SWP2008s-01.MEDARCH.ORG
                cifs/VM-SWP2008s-02.MEDARCH.ORG
                cifs/engdm
                cifs/engdm.MEDARCH.ORG
                cifs/enterprise
                cifs/enterprise.wwmed.com
                cifs/ntap-prov
                cifs/ntap-prov.MEDARCH.ORG
                cifs/ntap820
                cifs/ntap820.MEDARCH.ORG
Description:           insurance-claim records
State:                 Enabled
SMB2 Protocol:         Not Available
Signatures:            Enabled
WINS Name Encoding:    ISO-8859-1

Exports for Namespace: insur

  Share Name                            Volume Path                           State
  ---------------------------------- - -------------------------------------- -------
  CLAIMS                                /claims                               Online
  SPECS                                 /claims/specs                         Online
  STATS                                /claims/stats                          Online

Exports for Namespace: medarcv

  Share Name                           Volume Path                            State
  ---------------------------------- - -------------------------------------- -------
  ARCHIVES                             /rcrds                                 Online
  bulkstorage                          /rcrds                                 Online
  CELEBS                             S /rcrds/VIP_wing                        Online
  chem_results                         /test_results                          Online
  labs                                 /lab_equipment                         Online
  MP3S                               S /rcrds/2011/mp3downloads               Online
  xraysScanners                        /lab_equipment                         Online
  Y2004                              S /rcrds/2004                            Online
  Y2005                              S /rcrds/2005                            Online
  Y2008                              S /rcrds/2008                            Online
  Y2010                              S /rcrds/2010                            Online

  S = filer-subshare export

bstnA>
```

Showing Details for a CIFS Service

You can add an optional argument, detailed, for details about each CIFS share in the service:

show cifs-service *fqdn* detailed

For example:

```
bstnA> show cifs-service ac1.MEDARCH.ORG detailed

Service Name:          ac1.MEDARCH.ORG
Domain Join:           Joined to MEDARCH.ORG
Account Name:          ac1$
Delegation:            Constrained, Any Protocol
Delegate To:           cifs/PV770N
                cifs/PV770N.MEDARCH.ORG
                cifs/VM-PV770N-01
                cifs/VM-PV770N-01.MEDARCH.ORG
                cifs/VM-SWP2003S1-5
                cifs/VM-SWP2003S1-5.MEDARCH.ORG
                cifs/VM-SWP2003S2-04
                cifs/VM-SWP2003s2-04.MEDARCH.ORG
                cifs/VM-SWP2008S-01
                cifs/VM-SWP2008S-02
                cifs/VM-SWP2008s-01.MEDARCH.ORG
                cifs/VM-SWP2008s-02.MEDARCH.ORG
                cifs/engdm
                cifs/engdm.MEDARCH.ORG
                cifs/enterprise
                cifs/enterprise.wwmed.com
                cifs/ntap-prov
                cifs/ntap-prov.MEDARCH.ORG
                cifs/ntap820
                cifs/ntap820.MEDARCH.ORG
Description:           insurance-claim records
State:                 Enabled
SMB2 Protocol:         Not Available
Signatures:            Enabled
WINS Name Encoding:    ISO-8859-1
```

```
Shares
------
  ARCHIVES
    Namespace        medarcv
    Volume path      /rcrds
    Description      2 year-old medical records
    Export state     Ready
    Path state       Online
    Filer-subshare   No
    Offline-access   None

  Y2005
    Namespace        medarcv
    Volume path      /rcrds/2005
    Description
    Export state     Ready
    Path state       Online
    Filer-subshare   Yes
    Offline-access   Manual

  labs
    Namespace        medarcv
    Volume path      /lab_equipment
    Description      lab equipment
    Export state     Ready
    Path state       Online
    Filer-subshare   No
    Offline-access   Manual

  bulkstorage
    Namespace        medarcv
    Volume path      /rcrds
    Description      big share, now merged thru ARX
    Export state     Ready
    Path state       Online
    Filer-subshare   No
    Offline-access   Manual

  xraysScanners
    Namespace        medarcv
    Volume path      /lab_equipment
    Description      scanners and xray machines
    Export state     Ready
    Path state       Online
    Filer-subshare   No
    Offline-access   Manual

  chem_results
    Namespace        medarcv
    Volume path      /test_results
    Description      chem-lab records
    Export state     Ready
    Path state       Online
    Filer-subshare   No
    Offline-access   Manual

  E$
    Namespace        medarcv
    Volume path      /acopia$ns3
    Description      Managed volumes in namespace 'medarcv'
    Export state     Ready
    Path state       Online
    Filer-subshare   No
    Offline-access   Manual

  Z$
    Namespace        medarcv
    Volume path      /test_results
    Description      Volume '/test_results' in namespace 'medarcv'
    Export state     Ready
    Path state       Online
    Filer-subshare   No
    Offline-access   Manual

  CLAIMS
    Namespace        insur
    Volume path      /claims
    Description      insurance claims
    Export state     Ready
    Path state       Online
    Filer-subshare   No
    Offline-access   Manual

  CELEBS
    Namespace        medarcv
    Volume path      /rcrds/VIP_wing
    Description
    Export state     Ready
    Path state       Online
    Filer-subshare   Yes (hidden)
    Offline-access   Manual

  MP3S
    Namespace        medarcv
    Volume path      /rcrds/2011/mp3downloads
    Description
    Export state     Ready
    Path state       Online
    Filer-subshare   Yes
    Offline-access   Manual

  Y2004
    Namespace        medarcv
    Volume path      /rcrds/2004
    Description
    Export state     Ready
    Path state       Online
    Filer-subshare   Yes
    Offline-access   Manual

  Y2010
    Namespace        medarcv
    Volume path      /rcrds/2010
    Description
    Export state     Ready
    Path state       Online
    Filer-subshare   Yes
    Offline-access   Manual

  SPECS
    Namespace        insur
    Volume path      /claims/specs
    Description
    Export state     Ready
```

```
      Path state       Online
      Filer-subshare   No
      Offline-access   Manual

    STATS
      Namespace        insur
      Volume path      /claims/stats
      Description      claim stats
      Export state     Ready
      Path state       Online
      Filer-subshare   No
      Offline-access   Manual

    F$
      Namespace        insur
      Volume path      /acopia$ns4
      Description      Managed volumes in namespace 'insur'
      Export state     Ready
      Path state       Online
      Filer-subshare   No
      Offline-access   Manual

    Y2008
      Namespace        medarcv
      Volume path      /rcrds/2008
      Description
      Export state     Ready
      Path state       Online
      Filer-subshare   Yes
      Offline-access   Manual

bstnA>
```

Showing All CIFS Services

To show all CIFS front-end services, use show cifs-service all:

show cifs-service all [detailed]

where **detailed** (optional) adds details to the CIFS shares.

For example, this shows a summary view of every CIFS service on the ARX:

```
bstnA# show cifs-service all

Service Name:          ac1.MEDARCH.ORG
Domain Join:           Joined to MEDARCH.ORG
Account Name:          ac1$
Delegation:            Constrained, Any Protocol
Delegate To:           cifs/PV770N
                cifs/PV770N.MEDARCH.ORG
                cifs/VM-PV770N-01
                cifs/VM-PV770N-01.MEDARCH.ORG
                cifs/VM-SWP2003S1-5
                cifs/VM-SWP2003S1-5.MEDARCH.ORG
                cifs/VM-SWP2003S2-04
                cifs/VM-SWP2003s2-04.MEDARCH.ORG
                cifs/VM-SWP2008S-01
                cifs/VM-SWP2008S-02
                cifs/VM-SWP2008s-01.MEDARCH.ORG
                cifs/VM-SWP2008s-02.MEDARCH.ORG
                cifs/engdm
                cifs/engdm.MEDARCH.ORG
                cifs/enterprise
                cifs/enterprise.wwmed.com
                cifs/ntap-prov
                cifs/ntap-prov.MEDARCH.ORG
                cifs/ntap820
                cifs/ntap820.MEDARCH.ORG
Description:           insurance-claim records
State:                 Enabled
SMB2 Protocol:         Not Available
Signatures:            Enabled
WINS Name Encoding:    ISO-8859-1

Exports for Namespace: insur

  Share Name                           Volume Path                            State
  ---------------------------------- - -------------------------------------- -------
  CLAIMS                               /claims                                Online
  SPECS                                /claims/specs                          Online
  STATS                                /claims/stats                          Online

Exports for Namespace: medarcv

  Share Name                           Volume Path                            State
  ---------------------------------- - -------------------------------------- -------
  ARCHIVES                             /rcrds                                 Online
  bulkstorage                          /rcrds                                 Online
  CELEBS                             S /rcrds/VIP_wing                        Online
  chem_results                         /test_results                          Online
  labs                                 /lab_equipment                         Online
  MP3S                               S /rcrds/2011/mp3downloads               Online
  xraysScanners                        /lab_equipment                         Online
  Y2004                              S /rcrds/2004                            Online
  Y2005                              S /rcrds/2005                            Online
  Y2010                              S /rcrds/2010                            Online

  S = filer-subshare export


Service Name:          labtests.MEDARCH.ORG
Domain Join:           Joined to MEDARCH.ORG
Account Name:          labtests$
Delegation:            Constrained, Any Protocol
Delegate To:           cifs/VM-SMB2K8R2-01
                cifs/VM-SMB2K8R2-01.MEDARCH.ORG
                cifs/VM-SMB2K8R2-02
                cifs/VM-SMB2K8R2-02.MEDARCH.ORG
                cifs/VM-SWP2008S-01
                cifs/VM-SWP2008S-02
                cifs/VM-SWP2008s-01.MEDARCH.ORG
                cifs/VM-SWP2008s-02.MEDARCH.ORG
Description:           tests and records for lab animals
State:                 Enabled
SMB2 Protocol:         Available (2.002)
Signatures:            Not Enabled
WINS Name Encoding:    ISO-8859-1

Exports for Namespace: labResearch

  Share Name                           Volume Path                            State
  ---------------------------------- - -------------------------------------- -------
  MICE                               /labMice                             Online
  RATS                               /labRats                             Online

bstnA#
```

Showing Subshares

The show cifs-service subshares command shows a table of front-end subshares and the filer subshares behind them:

show cifs-service subshares {all | *fqdn*}

where **all** | ***fqdn*** is a required choice:

all shows the subshares for all CIFS services, and

fqdn (1-128 characters) identifies a single CIFS service. This is the fully-qualified domain name (for example, www.startup.com) for the CIFS service's global server.

For example, the following command shows the subshares behind the "ac1.medarch.org" service:

```
bstnA# show cifs-service subshares ac1.MEDARCH.ORG

Service:          ac1.MEDARCH.ORG

Export               Filer                Share                Subshare             State
-------------------- -------------------- -------------------- -------------------- --------
Y2005                fs4                  prescriptions        Y2005                Ready
                     fs1                  histories            Y2005                Ready
                     fs2                  bulkstorage          Y2005                Ready
CELEBS               fs4                  prescriptions        CELEBS$              Ready
                     fs1                  histories            CELEBS$              Ready
                     fs2                  bulkstorage          CELEBS$              Ready
MP3S                 fs4                  prescriptions        MP3S                 Ready
                     fs1                  histories            MP3S                 Ready
                     fs2                  bulkstorage          MP3S                 Ready
Y2004                fs4                  prescriptions        Y2004                Ready
                     fs1                  histories            Y2004                Ready
                     fs2                  bulkstorage          Y2004                Ready
Y2010                fs4                  prescriptions        Y2010                Ready
                     fs1                  histories            Y2010                Ready
                     fs2                  bulkstorage          Y2010                Ready
Y2008                fs4                  prescriptions        Y2008                Ready
                     fs1                  histories            Y2008                Ready
                     fs2                  bulkstorage          Y2008                Ready

bstnA#
```

Sample - Configuring a CIFS Front-End Service

The following command sequence sets up CIFS service on a global server called “ac1.medarch.org:”

```
bstnA(gbl)# cifs ac1.medarch.org
bstnA(gbl-cifs[ac1.medarch.org])# show global-config namespace medarcv
;====================== namespace managed volumes =======================
namespace medarcv
  protocol cifs
  cifs authentication kerberos

...

  volume /rcrds
    filer-subshares
    modify
    reimport-modify

...

  exit

bstnA(gbl-cifs[ac1.medarch.org])# export medarcv /rcrds as ARCHIVES description “2-year-old
medical records”
bstnA(gbl-cifs[ac1.medarch.org])# description “medical histories and records”
bstnA(gbl-cifs[ac1.medarch.org])# domain-join MEDARCH.ORG
Username: acoadmin
Password: ******

% INFO: Service 'ac1' successfully joined the domain using a pre-created computer account.


% INFO: Service 'ac1' joined the domain with delegation type set to Constrained. Only certain
selected services are allowed to be delegated on 'bgh'. The Active Directory administrator must
configure the services 'bgh' may delegate to, which are the filer(s) this CIFS service
virtualizes.


bstnA(gbl-cifs[ac1.medarch.org])# enable
bstnA(gbl-cifs[ac1.medarch.org])# exit
bstnA(gbl)# ...
```

Removing a CIFS Service

You can remove a CIFS service from a global server to both disable the service and remove its configuration. Use the no form of the cifs command to remove an CIFS-service configuration from a global server:

```
no cifs fqdn
```

where ***fqdn*** is the fully-qualified domain name (for example, “www.organization.org”) for the global server.

The CLI prompts for confirmation before removing the service; enter **yes** to proceed.

For example, the following command sequence removes the CIFS-service offering for ac1.medarch.org:

```
bstnA(gbl)# no cifs ac1.medarch.org
Delete CIFS service on 'ac1.medarch.org'? [yes/no] yes
bstnA(gbl)#
```

Verifying That a Service Can Support SMB2

SMB2 is a newer version of the original CIFS protocol, widely used in network-storage equipment. All of the back-end filers behind a CIFS service must support SMB2 in order for the CIFS service to support SMB2. An initial CIFS connection to a front-end service does not identify the client's desired namespace(s) or volume(s), so SMB2 is required at all of them.

Consulting the Compatibility Matrix

We recommend using only SMB2 filers that have been tested for SMB2 support at F5. The *F5 Data Solutions Compatibility Matrix* (included with this doc set) lists all filers that have been fully qualified for SMB2 support behind the ARX.

You can use the command described in the next subsection to get a full list of the filers behind the service, and to probe the filers for SMB2 support, but we recommend that you only use filers that are listed in the Compatibility Matrix.

Checking for SMB2 Support on the Back-End Filers

To check all of the back-end filers behind a given CIFS service for SMB2 support, use the show cifs-service filer-capabilities command:

```
show cifs-service filer-capabilities fqdn
```

> where ***fqdn*** (1-128 characters) is the fully-qualified domain name (for example, www.company.com) for the CIFS service's global server.

This displays information collected when each of the shares was enabled. For instructions on enabling a share, recall *Enabling the Share*, on page 8-16 and/or *Enabling the Share*, on page 9-45. The shares must be enabled in an ARX volume before the information in this output is complete.

The output shows the SMB and SMB2 parameters for every filer behind the CIFS service, the merged capabilities of all of them, and the final capabilities of the service. The Merged capabilities are the capabilities that all of the filers can support, and the Service capabilities are all the capabilities from the Merged list that are supportable by the ARX. The back-end filer(s) show "SMB2: not available" if they do not support SMB2; if this appears for any of the filers, the service cannot support SMB2.

For example, the following command sequence finds that SMB2 is supportable for the "labtests.medarch.org" service:

```
bstnA(gbl)# show cifs-service filer-capabilities labtests.MEDARCH.ORG

Capabilities of CIFS/SMB2 filers used by service labtests.MEDARCH.ORG

Filer fs3 [192.168.25.28]
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2.002: Max Read 65536, Max Write 65536, Max Transact 65536
```

```
Filer fs4 [192.168.25.29]
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2.002: Max Read 65536, Max Write 65536, Max Transact 65536

Filer fs6 [192.168.25.30]
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2.1: Max Read 1048576, Max Write 1048576, Max Transact 1048576
       Leasing, Large MTU

Filer fs7 [192.168.25.41]
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2.1: Max Read 1048576, Max Write 1048576, Max Transact 1048576
       Leasing, Large MTU

Merged capabilities
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2.002: Max Read 65536, Max Write 65536, Max Transact 65536

Service capabilities
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2.002: Max Read 65536, Max Write 65536, Max Transact 65536

bstnA(gbl)# ...
```

Confirming SMB2 Support for the Front-End Service

There may be a slight lag between the time the last back-end server becomes SMB2-capable and the time that the front-end CIFS service can offer SMB2. To confirm that the service is ready, also run the show cifs-service command and verify that the SMB2 Protocol field reports that SMB2 is “Available.” For example:

```
bstnA(gbl)# show cifs-service labtests.MEDARCH.ORG

Service Name:          labtests.MEDARCH.ORG
Domain Join:           Joined to MEDARCH.ORG
Account Name:          labtests$
Delegation:            Constrained, Any Protocol
Delegate To:           cifs/VM-SMB2K8R2-01
                cifs/VM-SMB2K8R2-01.MEDARCH.ORG
                cifs/VM-SMB2K8R2-02
                cifs/VM-SMB2K8R2-02.MEDARCH.ORG
                cifs/VM-SWP2008S-01
                cifs/VM-SWP2008S-02
                cifs/VM-SWP2008s-01.MEDARCH.ORG
                cifs/VM-SWP2008s-02.MEDARCH.ORG
Description:           tests and records for lab animals
State:                 Enabled
SMB2 Protocol:         Available (2.002)
Signatures:            Not Enabled
WINS Name Encoding:    ISO-8859-1
```

```
Exports for Namespace: labResearch

  Share Name                           Volume Path                           State
  ---------------------------------- - ------------------------------------- -------
  MICE                                 /labMice                              Online
  RATS                                 /labRats                              Online

bstnA(gbl)# ...
```

When SMB2 is Not Supported

As another example, the following command sequence finds that SMB2 is unsupportable for several filers behind the “ac1.medarch.org” service, and is therefore unavailable for the service as a whole:

```
bstnA(gbl)# show cifs-service filer-capabilities ac1.MEDARCH.ORG

Capabilities of CIFS/SMB2 filers used by service ac1.MEDARCH.ORG

Filer fs1 [192.168.25.20]
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2: not available

Filer fs2 [192.168.25.27]
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2: not available

Filer fs3 [192.168.25.28]
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2.002: Max Read 65536, Max Write 65536, Max Transact 65536

Filer fs4 [192.168.25.29]
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2.002: Max Read 65536, Max Write 65536, Max Transact 65536

Filer fs5 [192.168.25.71]
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write, Info Passthru
  SMB2: not available

Filer nas1 [192.168.25.21]
  SMB: Max Buffer 33028
       Large Files, Large Read, Large Write
  SMB2: not available

Filer nas10 [192.168.25.49]
  SMB: Max Buffer 33028
       Large Files, Large Read, Large Write
  SMB2.002: Max Read 65536, Max Write 65536, Max Transact 65536

Filer nasE1 [192.168.25.51]
  SMB: Max Buffer 65535
       Large Files, Large Read, Large Write, Info Passthru
  SMB2.002: Max Read 65536, Max Write 65536, Max Transact 65536

Merged capabilities
```

```
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write
  SMB2: not available

Service capabilities
  SMB: Max Buffer 16644
       Large Files, Large Read, Large Write
  SMB2: not available

bstnA(gbl)# ...
```

Upgrading the Service to Support SMB2

All back-end shares behind a CIFS service must support SMB2 for the CIFS service to support it. Remove all volume shares that do not support SMB2. Recall *Removing a Direct Share*, on page 8-17 for instructions on removing a direct-volume share. The *ARX CLI Maintenance Guide* explains the more-difficult process of removing a managed-volume share: see *Removing an Imported Share*, on page 10-18 of that manual.

If you replace these shares with new ones that support SMB2 (recall *Adding a Share*, on page 8-8 and/or *Adding a Share*, on page 9-33), allow at least 15 minutes to pass after adding the new shares before trying any SMB2 connections. The volume software probes its back-end shares periodically for SMB2 capabilities, and the CIFS service does not support SMB2 until the probe is complete at all of its filers.

Removing All of a Volume's Front-End Exports

You can use a single command to remove all of the front-end exports, NFS and/or CIFS, for a given volume. This is convenient for a volume that has been exported through multiple global servers and front-end services. The remove namespace command has been described in previous chapters for removing a volume (see *Removing a Direct Volume*, on page 8-31 or *Removing a Managed Volume*, on page 9-68) or an entire namespace (*Removing a Namespace*, on page 7-28); add the optional exports-only keyword to remove only the front-end exports:

```
remove namespace name volume volume exports-only [timeout seconds]
[sync]
```

where:

name (1-30 characters) is the name of the namespace,

volume (optional, 1-1024 characters) is the path name of the volume,

exports-only is the option to remove only the front-end exports for the volume,

seconds (optional, 300-10,000) sets a time limit on each of the removal's component operations, and

sync (optional) waits for the removal to finish before returning. With this option, the CLI lists the configuration objects as it removes them.

The CLI prompts for confirmation before removing the exports. Enter **yes** to continue. This operation generates a report named "removeNs_*namespace_date*.rpt" if you omit the sync option.

For example, this command sequence exits to priv-exec mode and then synchronously removes all front-end exports from the "insur_bkup~/insurShdw" volume:

```
prtlndA(gbl)# end
prtlndA# remove namespace insur_bkup volume /insurShdw exports-only sync

Remove exports for namespace 'insur_bkup', volume '/insurShdw'? [yes/no] yes
% INFO: Removing service configuration for namespace insur_bkup
% INFO: Removing NFS services for namespace insur_bkup
% INFO: Removing CIFS services for namespace insur_bkup
% INFO: no export insur_bkup /insurShdw as CLAIMS_BKUP
prtlndA# ...
```

Showing All Front-End Services

Front-end services are identified by the FQDN of their respective global servers. Use the show global service command to show all front-end services configured on the ARX:

```
show global service
```

For example:

```
bstnA(gbl)# show global service

Domain Name                 Service     State
--------------------------------------------------
acopiaFiler                 NFS         Enabled
ac1.MEDARCH.ORG             NFS         Enabled
ac1.MEDARCH.ORG             CIFS        EnabledbstnA(gbl)# ...
```

Showing Front-End Services for One Global-Server

To see the front-end services for one global server, identify a particular global server with the show global service command:

```
show global service fqdn
```

where ***fqdn*** (1-128 characters) is the fully-qualified domain name (for example, www.company.com) for the global server.

For example, the following command shows the front-end services for the global server at “www.nemed.com:”

```
bstnA(gbl)# show global service www.nemed.com

Domain Name                 Service     State
--------------------------------------------------
www.nemed.com               NFS         Enabled

bstnA(gbl)# ...
```

Showing Front-End Services per Virtual-Server

You can show the front-end services running at each virtual server, with the VIP and current health of each service. To see the front-end services grouped by their virtual servers, use the show virtual service command:

```
show virtual service
```

For example:

```
bstnA(gbl)# show virtual service
Switch:  bstnA
------------------------

  Global Server                           Virtual IP Address   Service    State
  ------------------------------------------------------------------------------------
  acopiaFiler                             192.168.25.12        NFS        Ready
  ac1.MEDARCH.ORG                         192.168.25.15        NFS        Ready
  ac1.MEDARCH.ORG                         192.168.25.15        CIFS       Ready

bstnA(gbl)# ...
```

In a redundant pair, this shows both peers and the virtual services that are running on each. For example:

```
prtlndA# show virtual service
Switch:  prtlndA
------------------------

  Global Server                              Virtual IP Address   Service    State
  ---------------------------------------------------------------------------------------
  www.nemed.com                              192.168.74.91        NFS        Ready
  insurBkup.MEDARCH.ORG                      192.168.74.92        CIFS       Ready

Switch:  prtlndB
------------------------

  Global Server                              Virtual IP Address   Service    State
  ---------------------------------------------------------------------------------------


prtlndA# ...
```

Showing the Services at the Redundant Peer

To focus on one peer in the redundant pair, identify the peer switch at the end of the command:

show virtual service *peer-name*

where ***peer-name*** (1-128 characters) identifies the peer by its hostname.

For example, this shows the services running on “prtlndA” from its redundant peer, “prtlndB:”

```
prtlndB# show virtual service prtlndA
Switch:  prtlndA
------------------------

  Global Server                              Virtual IP Address   Service    State
  ---------------------------------------------------------------------------------------
  www.nemed.com                              192.168.74.91        NFS        Ready
  insurBkup.MEDARCH.ORG                      192.168.74.92        CIFS       Ready


prtlndB# ...
```

Showing Server Maps

You can show the map between front-end services and the back-end servers behind them. From any mode, use the show server-mapping command:

```
show server-mapping
```

This displays a two-column table, where the left column shows the client-side view and the right column shows the server side. Each front-end export has its own listing of back-end filers. For example, the following command shows all front-end exports on the “bstnA” switch with their backing filers:

```
bstnA(gbl)# show server-mapping

Virtual Server               Namespace/Volume
   Virtual Path                 Physical Server
-----------------------------------------------------------------------
192.168.25.12:/vol           medco:/vol

   vol1/corp                     nas1:/vol/vol2/shr
   vol1/notes                    nas1:/vol/vol2/notes
   vol2                          nas3:/exports/data
   vol3/mtgMinutes               nas2:/vol/datavol1/direct/mtgs
   vol3/sales                    nas2:/vol/datavol1/direct/export


192.168.25.15:/acct          wwmed:/acct

                                 das1:/exports/budget
                                 das3:/exports/acct2
                                 das7:/lhome/it5
                                 das8:/work1/accting
                                 nas1:/vol/vol2/meta1*


192.168.25.15:/acct/wksheets wwmed:/acct

                                 das1:/exports/budget
                                 das3:/exports/acct2
                                 das7:/lhome/it5
                                 das8:/work1/accting
                                 nas1:/vol/vol2/meta1*


192.168.25.15:/claims        insur:/claims

                                 nas1:/vol/vol2/insurance
                                 nas1:/vol/vol2/meta2*
                                 nasE1:/patient_records


\\192.168.25.15\ARCHIVES     medarcv:/rcrds

                                 \\fs1\histories
                                 \\fs2\bulkstorage
                                 \\fs4\prescriptions
                                 nas1:/vol/vol2/meta3*


\\192.168.25.15\bulkstorage  medarcv:/rcrds
```

```
                                 \\fs1\histories
                                 \\fs2\bulkstorage
                                 \\fs4\prescriptions
                                 nas1:/vol/vol2/meta3*


\\192.168.25.15\CELEBS       medarcv:/rcrds

                                 \\fs1\histories
                                 \\fs2\bulkstorage
                                 \\fs4\prescriptions
                                 nas1:/vol/vol2/meta3*


\\192.168.25.15\chem_results medarcv:/test_results

   2005charts                    ** medarcv:/rcrds/2005
   chemLab                       \\fs1\chem_results/.
   hematologyLab                 \\fs3\hematology_results/.


\\192.168.25.15\CLAIMS       insur:/claims

                                 nas1:/vol/vol2/meta2*
                                 \\nas1\insurance
                                 \\nasE1\patient_records


\\192.168.25.15\labs         medarcv:/lab_equipment

                                 \\fs2\backlot_records
                                 \\fs5\xraysScanners
                                 nas1:/vol/vol2/meta6*
                                 \\nas10\equipment
                                 \\nas10\for_lease
                                 \\nas11\equipBkup
                                 \\nas11\leasedBkup


\\192.168.25.15\MP3S         medarcv:/rcrds

                                 \\fs1\histories
                                 \\fs2\bulkstorage
                                 \\fs4\prescriptions
                                 nas1:/vol/vol2/meta3*


\\192.168.25.15\SPECS        insur:/claims

                                 nas1:/vol/vol2/meta2*
                                 \\nas1\insurance
                                 \\nasE1\patient_records


\\192.168.25.15\STATS        insur:/claims

                                 nas1:/vol/vol2/meta2*
                                 \\nas1\insurance
                                 \\nasE1\patient_records
```

```
\\192.168.25.15\xraysScannersmedarcv:/lab_equipment

                                 \\fs2\backlot_records
                                 \\fs5\xraysScanners
                                 nas1:/vol/vol2/meta6*
                                 \\nas10\equipment
                                 \\nas10\for_lease
                                 \\nas11\equipBkup
                                 \\nas11\leasedBkup


\\192.168.25.15\Y2004        medarcv:/rcrds

                                 \\fs1\histories
                                 \\fs2\bulkstorage
                                 \\fs4\prescriptions
                                 nas1:/vol/vol2/meta3*


\\192.168.25.15\Y2005        medarcv:/rcrds

                                 \\fs1\histories
                                 \\fs2\bulkstorage
                                 \\fs4\prescriptions
                                 nas1:/vol/vol2/meta3*


\\192.168.25.15\Y2010        medarcv:/rcrds

                                 \\fs1\histories
                                 \\fs2\bulkstorage
                                 \\fs4\prescriptions
                                 nas1:/vol/vol2/meta3*


\\192.168.25.16\MICE         labResearch:/labMice

                                 \\fs3\mice_lab_results
                                 \\fs4\white_mice
                                 nas1:/vol/vol2/meta8*


\\192.168.25.16\RATS         labResearch:/labRats

                                 \\fs6\labRats
                                 \\fs7\lrTests
                                 nas1:/vol/vol2/meta9*



Where * denotes metadata only physical server.
Where ** denotes a direct volume mapped to a namespace.
bstnA(gbl)# ...
```

Showing the Servers Behind One Namespace

You can also show the physical servers behind one namespace:

show server-mapping namespace *name* [ip-addresses]

where

name (1-30 characters) identifies the namespace, and

ip-addresses (optional) was explained earlier.

If multiple virtual servers provide service for the namespace, this command shows all of them.

For example, the following command shows the filers behind the “wwmed” namespace. This shows IP addresses instead of external-filer names:

```
bstnA(gbl)# show server-mapping namespace wwmed ip-addresses

Virtual Server               Namespace/Volume
   Virtual Path                 Physical Server
-----------------------------------------------------------------------
192.168.25.15:/acct          wwmed:/acct

                                 192.168.25.19:/exports/budget
                                 192.168.25.23:/exports/acct2
                                 192.168.25.24:/lhome/it5
                                 192.168.25.25:/work1/accting
                                 192.168.25.47:/exports/meta7*


192.168.25.15:/acct/wksheets wwmed:/acct

                                 192.168.25.19:/exports/budget
                                 192.168.25.23:/exports/acct2
                                 192.168.25.24:/lhome/it5
                                 192.168.25.25:/work1/accting
                                 192.168.25.47:/exports/meta7*



Where * denotes metadata only physical server.
Where ** denotes a direct volume mapped to a namespace.
bstnA(gbl)# ...
```

Showing Server Status

The status keyword shows high-level status for each of the servers. The status is shown for each virtual server as well as each of the shares behind it. This provides a high-level view of each virtual server’s health:

show server-mapping status [ip-addresses]

where **ip-addresses** (optional) shows filer IPs instead of their external-filer names, as explained above.

To examine the configuration and status for a each share, use the show namespace command. See *Showing Namespace Details*, on page 7-4.

For example, the following command shows that all virtual servers are ready and all shares are online:

```
bstnA(gbl)# show server-mapping status

Virtual Server
        Physical Server                                             Status
---------------------------------------------------------------    --------
192.168.25.12:/vol                                                  Ready

        nas1:/vol/vol2/shr                                          Online
        nas2:/vol/datavol1/direct/mtgs                              Online
        nas1:/vol/vol2/notes                                        Online
        nas2:/vol/datavol1/direct/export                            Online
        nas3:/exports/data                                          Online

192.168.25.15:/acct                                                 Ready

        das1:/exports/budget                                        Online
        das8:/work1/accting                                         Online
        das3:/exports/acct2                                         Online
        das7:/lhome/it5                                             Online

192.168.25.15:/acct/wksheets                                        Ready

        das1:/exports/budget                                        Online
        das8:/work1/accting                                         Online
        das3:/exports/acct2                                         Online
        das7:/lhome/it5                                             Online

192.168.25.15:/claims                                               Ready

        nas1:/vol/vol2/insurance                                    Online
        nasE1:/patient_records                                      Online

\\192.168.25.15\CLAIMS                                              Ready

        \\nas1\insurance                                            Online
        \\nasE1\patient_records                                     Online

\\192.168.25.15\SPECS                                               Ready

        \\nas1\insurance                                            Online
        \\nasE1\patient_records                                     Online

\\192.168.25.15\STATS                                               Ready

        \\nas1\insurance                                            Online
        \\nasE1\patient_records                                     Online

\\192.168.25.15\labs                                                Ready

        \\nas10\equipment                                           Online
        \\nas10\for_lease                                           Online
        \\fs2\backlot_records                                       Online
        \\fs5\xraysScanners                                         Online
        \\nas11\equipBkup                                           Online
        \\nas11\leasedBkup                                          Online

\\192.168.25.15\xraysScanners                                       Ready

        \\nas10\equipment                                           Online
        \\nas10\for_lease                                           Online
        \\fs2\backlot_records                                       Online
        \\fs5\xraysScanners                                         Online
        \\nas11\equipBkup                                           Online
        \\nas11\leasedBkup                                          Online

\\192.168.25.15\ARCHIVES                                            Ready

        \\fs4\prescriptions                                         Online
        \\fs1\histories                                             Online
        \\fs2\bulkstorage                                           Online

\\192.168.25.15\bulkstorage                                         Ready

        \\fs4\prescriptions                                         Online
        \\fs1\histories                                             Online
        \\fs2\bulkstorage                                           Online

\\192.168.25.15\Y2004                                               Ready

        \\fs4\prescriptions                                         Online
        \\fs1\histories                                             Online
        \\fs2\bulkstorage                                           Online

\\192.168.25.15\Y2005                                               Ready

        \\fs4\prescriptions                                         Online
        \\fs1\histories                                             Online
        \\fs2\bulkstorage                                           Online

\\192.168.25.15\Y2010                                               Ready

        \\fs4\prescriptions                                         Online
        \\fs1\histories                                             Online
        \\fs2\bulkstorage                                           Online

\\192.168.25.15\MP3S                                                Ready

        \\fs4\prescriptions                                         Online
        \\fs1\histories                                             Online
        \\fs2\bulkstorage                                           Online

\\192.168.25.15\CELEBS                                              Ready

        \\fs4\prescriptions                                         Online
        \\fs1\histories                                             Online
        \\fs2\bulkstorage                                           Online

\\192.168.25.15\chem_results                                        Ready

        ** medarcv:/rcrds/2005                                      Online
        \\fs1\chem_results/.                                        Online
        \\fs3\hematology_results/.                                  Online

\\192.168.25.16\MICE                                                Ready

        \\fs3\mice_lab_results                                      Online
        \\fs4\white_mice                                            Online

\\192.168.25.16\RATS                                                Ready

        \\fs6\labRats                                               Online
        \\fs7\lrTests                                              Online

bstnA(gbl)# ...
```

12

Creating a Policy Schedule

Overview

A *schedule* is a configuration object that you can apply to one or more policy rules. You must create a schedule in the ARX configuration before any rule can use it. You can skip this chapter if you do not plan to put any rules on a schedule.

From gbl mode, use the schedule command to create a policy schedule:

```
schedule name
```

where ***name*** (1-64 characters) is the name that you assign to the schedule,

For example, the following command creates a schedule named “hourly:”

```
bstnA(gbl)# schedule hourly
bstnA(gbl-schedule[hourly])# ...
```

The remaining sections describe your options for setting up the schedule in gbl-schedule mode.

Setting the Interval

The next step in creating a schedule is to set the schedule’s interval (such as “every 5 days” or “every 3 hours”). From gbl-schedule mode, use the every command to set the schedule’s interval:

```
every number {minutes|hours|days|weeks|months|quarters|years}
```

where

number is an integer and

minutes|hours|days|weeks|months|quarters|years is a required choice.

For example, **every 15 minutes**, **every 3 weeks**, **every 4 days**, or **every 1 years**.

The following sample-command sequence configures the “hourly” schedule to fire once per hour:

```
bstnA(gbl)# schedule hourly
bstnA(gbl-schedule[hourly])# every 1 hours
bstnA(gbl-schedule[hourly])# ...
```

Setting a Monthly Interval

You have four options for setting a monthly interval:

```
every number months
```

or

```
every day-list day-of-month[,day-of-month]+
```

or

```
every first day-of-week
```

or

```
every last day-of-week
```

where you can choose between four formats:

- **every *number* months** was described above. This sets an interval of one or more months.
- **every day-list *day-of-month+*** (1-31) specifies a list of days to run the schedule in each month. For example, **every day-list 1,15,31** creates a schedule that fires on three days of every month: the 1st, the 15th, and the 31st. For shorter months, the policy engine interprets a "31" as the last day of the month.
- **every first *day-of-week***, where the *day-of-week* is the full name of the day in lower case. For example, **every first monday** or **every first sunday**. This creates a schedule that runs once per month on the specified day.
- **every last *day-of-week*** is a once-per-month schedule that runs on the last specified day. For example, **every last wednesday**.

The following sample-command sequence configures the "twiceMonthly" schedule to fire twice each month:

```
bstnA(gbl)# schedule twiceMonthly
bstnA(gbl-schedule[twiceMonthly])# every day-list 1,15
bstnA(gbl-schedule[twiceMonthly])# ...
```

Setting a Weekly Interval, Possibly with Multiple Days

The next step in creating a schedule is to set the schedule's interval (such as "every 5 days," "every 3 hours," or "every Monday"). From gbl-schedule mode, use the every command to set the schedule's interval:

```
every number weeks
```

or

```
every day-of-week+
```

where you can choose between two formats:

- **every *number* weeks** was described earlier. This chooses some multiple of weeks, such as **every 3 weeks**.

- **every *day-of-week***, where the *day-of-week* is the full name of the day in lower case. For example, **every monday**, **every wednesday**, or **every sunday**. You can enter more than one to run the schedule more than once per week; for example, **every monday wednesday friday** runs three times per week.

The following sample-command sequence configures the “hourly” schedule to fire once per hour:

```
bstnA(gbl)# schedule hourly
bstnA(gbl-schedule[hourly])# every 1 hours
bstnA(gbl-schedule[hourly])# ...
```

Setting a Daily Interval, Possibly with Multiple Times

The following daily intervals are also available:

every *number* days

or

every hour-list *hour*:*minute*[,*hour*:*minute*]+

where you can choose between two formats:

- **every *number* days** was described earlier. This chooses some multiple of weeks, such as **every 5 days**.
- **every hour-list *hour*:*minute*** specifies a single time in the day when the schedule fires. You can list multiple times, separated with commas, to create a schedule that fires more than once per day. For example, **every hour-list 6:00,12:00,18:00** fires at 6 AM, noon, and 6 PM every day.

The following sample-command sequence configures the “twiceDaily” schedule to fire at 1 AM and 9 PM each day:

```
bstnA(gbl)# schedule twiceDaily
bstnA(gbl-schedule[twiceDaily])# every hour-list 01:00,21:00
bstnA(gbl-schedule[twiceDaily])# ...
```

Setting the Duration (optional)

The next, optional step in creating a schedule is to set a duration. A duration is not required for the schedule to function; this limits the amount of time that a rule can run each time the schedule fires. If you set a 5-minute duration for the schedule, each rule that uses the schedule has 5 minutes to finish processing every time it runs.

At the end of the duration, a rule with this schedule stops any of its volume scans and/or migrations. If a client changes a file so that it should migrate according to the rule, the policy engine caches the migration request. All migrations resume the next time the rule runs. On the other hand, the rule continues to direct *new* files and directories to their configured target shares; no migrations are necessary, so this is not controlled by the schedule or its duration.

From gbl-schedule mode, use the duration command to set the duration:

```
duration time-interval
```

where ***time-interval*** determines the duration of each scheduled event. Express the *time-interval* in *hh*:*mm*:*ss* format; for example, **duration 00:15:00** for 15 minutes, or **duration 06:00:00** for six hours.

For example, the following command sequence configures the “daily4am” schedule to run for 2 hours each time it fires:

```
bstnA(gbl)# schedule daily4am
bstnA(gbl-schedule[daily4am])# duration 02:00:00
bstnA(gbl-schedule[daily4am])# ...
```

Removing the Duration

The default is no limit on the duration of a rule execution. To remove the duration and return to this default, use no duration:

```
no duration
```

For example, the following command sequence removes the duration from the “hourly” schedule:

```
bstnA(gbl)# schedule hourly
bstnA(gbl-schedule[hourly])# no duration
bstnA(gbl-schedule[hourly])# ...
```

Setting the Start Time (optional)

A schedule's *start time* determines the start of each interval: if a daily schedule has a start time of 2:42 PM, the schedule will fire at 2:42 PM every day. By default, a schedule's start time is the time that it is configured. You can optionally use the start command to set a different start time and date:

```
start date:time
```

where the colon (**:**) separates the date from the time.

date defines the start date for the schedule. Use the format ***mm/dd/yyyy*** (for example, **04/01/2003** for April 1, 2003, or **11/20/2005** for November 20, 2005).

time defines the start time. Use the format ***HH*:*MM*:*SS*** (for example, **12:00:00** for noon, or **17:00:00** for 5 PM).

For example, the following command sequence sets the "daily4am" schedule to start at 4 AM on September 4, 2005:

```
bstnA(gbl)# schedule daily4am
bstnA(gbl-schedule[daily4am])# start 09/04/2005:04:00:00
bstnA(gbl-schedule[daily4am])# ...
```

Starting Now

Use the start command (without a date and time) to erase the previously-configured start time and start the schedule now:

```
start
```

This causes the schedule to fire immediately, invoking all rules that use it.

For example:

```
bstnA(gbl)# schedule hourly
bstnA(gbl-schedule[hourly])# start
bstnA(gbl-schedule[hourly])# ...
```

You can also use no start to reset the start time to "now," but no start does not cause the schedule to fire.

Adding a Description (optional)

You can add a description to the schedule for use in show commands. The description can differentiate the schedule from others in your show-command output. From gbl-schedule mode, use the description command to add a description:

```
description text
```

where ***text*** is 1-255 characters. Quote the text if it contains any spaces.

For example:

```
bstnA(gbl)# schedule daily4am
bstnA(gbl-schedule[daily4am])# description "two hours between 4 and 6 AM"
bstnA(gbl-schedule[daily4am])# ...
```

Removing the Description

The no description command removes any description from the current schedule:

```
no description
```

For example:

```
bstnA(gbl)# schedule hourly
bstnA(gbl-schedule[hourly])# no description
bstnA(gbl-schedule[hourly])# ...
```

Showing All Schedules

To list all schedules on the switch, use the show schedule command:

```
show schedule
```

This shows each schedule's configuration parameters as well as the times of the previous and next scheduled runs.

For example:

```
bstnA(gbl)# show schedule


 Schedule:         hourly
    Start Time:       08/18/2011 01:24:00 -0400
    Interval:         1 hours
    Status:           Running (runs in 00:03:34)

    Previous:
      Run Time:         08/18/2011 01:24:00 -0400 (Schedule Start)
      End Time:         N/A

    Next:
      Run Time:         08/18/2011 02:24:00 -0400
      End Time:         N/A


  Schedule:         daily4am
    Description:      two hours between 4 and 6 AM
    Start Time:       09/04/2005 04:00:00 -0400
    Stop Time:        01/07/2015 04:00:00 -0500 (Expires in 1238 d 02:39:34)
    Interval:         1 days
    Duration:         02:00:00
    Status:           Paused (runs in 01:39:34)

    Previous:
      Run Time:         08/17/2011 04:00:00 -0400
      End Time:         08/17/2011 06:00:00 -0400

    Next:
      Run Time:         08/18/2011 04:00:00 -0400
      End Time:         08/18/2011 06:00:00 -0400


  Schedule:         backupWindow
    Description:      regular backup times
    Start Time:       11/12/2006 13:00:00 -0500
    Interval:         1 days
    Duration:         04:00:00
    Status:           Paused (runs in 10:39:34)

    Previous:
      Run Time:         08/17/2011 13:00:00 -0400
      End Time:         08/17/2011 17:00:00 -0400

    Next:
      Run Time:         08/18/2011 13:00:00 -0400
      End Time:         08/18/2011 17:00:00 -0400


  Schedule:         weekly
```

```
    Start Time:       05/06/1995 02:00:00 -0400
    Interval:         1 weeks
    Status:           Running (runs in 1 d 23:39:34)

    Previous:
      Run Time:         08/13/2011 02:00:00 -0400
      End Time:         N/A

    Next:
      Run Time:         08/20/2011 02:00:00 -0400
      End Time:         N/A

bstnA(gbl)# ...
```

Showing One Schedule

To focus on a single schedule, add the desired schedule name to the command:

`show schedule` ***`name`***

where ***name*** (1-64 characters) identifies the schedule to show.

For example, this shows the “hourly” schedule:

```
bstnA(gbl)# show schedule hourly


 Schedule:         hourly
    Start Time:       08/18/2011 01:24:00 -0400
    Interval:         1 hours
    Status:           Running (runs in 00:37:07)

    Previous:
      Run Time:         08/18/2011 03:24:00 -0400
      End Time:         N/A

    Next:
      Run Time:         08/18/2011 04:24:00 -0400
      End Time:         N/A

bstnA(gbl)# ...
```

Removing a Schedule

Use the no form of the schedule command to remove a schedule from the switch configuration:

`no schedule` ***`name`***

where ***name*** (1-64 characters) identifies the schedule to be removed.

For example, the following command sequence shows all schedules and deletes the schedule named “backupWindow:”

```
bstnA(gbl)# show schedule
```

```
Schedule:         hourly
    Start Time:       08/18/2011 01:24:00 -0400
    Interval:         1 hours
    Status:           Running (runs in 00:03:34)

    Previous:
      Run Time:         08/18/2011 01:24:00 -0400 (Schedule Start)
      End Time:         N/A

    Next:
      Run Time:         08/18/2011 02:24:00 -0400
      End Time:         N/A


  Schedule:         daily4am
    Description:      two hours between 4 and 6 AM
    Start Time:       09/04/2005 04:00:00 -0400
    Stop Time:        01/07/2015 04:00:00 -0500 (Expires in 1238 d 02:39:34)
    Interval:         1 days
    Duration:         02:00:00
    Status:           Paused (runs in 01:39:34)

    Previous:
      Run Time:         08/17/2011 04:00:00 -0400
      End Time:         08/17/2011 06:00:00 -0400

    Next:
      Run Time:         08/18/2011 04:00:00 -0400
      End Time:         08/18/2011 06:00:00 -0400


  Schedule:         backupWindow
    Description:      regular backup times
    Start Time:       11/12/2006 13:00:00 -0500
    Interval:         1 days
    Duration:         04:00:00
    Status:           Paused (runs in 10:39:34)

    Previous:
      Run Time:         08/17/2011 13:00:00 -0400
      End Time:         08/17/2011 17:00:00 -0400

    Next:
      Run Time:         08/18/2011 13:00:00 -0400
      End Time:         08/18/2011 17:00:00 -0400


  Schedule:         weekly
    Start Time:       05/06/1995 02:00:00 -0400
    Interval:         1 weeks
    Status:           Running (runs in 1 d 23:39:34)

    Previous:
      Run Time:         08/13/2011 02:00:00 -0400
      End Time:         N/A

    Next:
      Run Time:         08/20/2011 02:00:00 -0400
      End Time:         N/A
bstnA(gbl)# no schedule backupWindow
bstnA(gbl)# ...
```

13

Grouping Files Into Filesets

Overview

This chapter pertains to managed volumes only. Direct volumes contain no metadata and do not support filesets.

A *fileset* is a group of files and/or directories to which you can apply replication and migration policies. You can configure filesets based on file name, directory path, size, age, and/or offline attribute. You can create complex filesets by joining multiple filesets in a union or taking the intersection of two or more filesets. This chapter explains how to create filesets, and the next chapter explains how to use policies to migrate them to your desired back-end filer(s) and possibly to implement tiered storage.

Grouping Files by Age

You can create filesets based on file age, or based on the most-recent time they changed. These filesets are a common component of tiered storage: a rule directs a fileset of “new” files to Tier-1 shares and another rule sends all older files to Tier-2 storage.

Each *age fileset* is “older-than” and/or “newer-than” a time of your choosing (6 hours, 3 weeks, or any other time frame). For example, you can create weekly filesets by creating one age fileset that is “newer-than 1 week,” another that is “older-than 1 week” but “newer-than 2 weeks,” and so on.

From gbl mode, use the policy-age-fileset command to create an age fileset:

```
policy-age-fileset name
```

where ***name*** (1-1024 characters) is the name that you assign to the fileset.

The CLI prompts for confirmation before creating the new fileset; enter **yes** to continue. This puts you into gbl-fs-age mode, where you must identify time frame for selecting files (newer-than and/or older-than a particular time). You can also select from several commands that alter the default selection criteria.

For example, the following command sequence creates an age fileset:

```
bstnA(gbl)# policy-age-fileset modThisMonth
This will create a new policy object.

Create object 'modThisMonth'? [yes/no] yes
bstnA(gbl-fs-age[modThisMonth])# ...
```

Selecting Files Based on their Ages

The next step in configuring an age fileset is to determine the age for its files. The age fileset defines a single age group for its files: older-than or newer-than a certain time interval or date. From gbl-fs-age mode, use the select-files command to identify the age group for the files:

```
select-files {older-than | newer-than} count {minutes | hours | days |
weeks | months | quarters | years}
```

where

older-than | newer-than is a required choice.

count (1-4,294,967,295) and **minutes | ... years** establishes a time frame (for example, **15 minutes**, **2 weeks**, or **1 quarter**).

You can run this command twice in the same fileset to choose files between two ages. For example, you can select-files older-than 2 months and select-files newer-than 3 months to get a set of files between 2 and 3 months old. You cannot run a select-files command that contradicts a previous one; if you already ran select-files newer-than 1 week, you cannot run select-files older-than 2 quarters.

For example, the following command set selects files for the "modThisMonth" fileset:

```
bstnA(gbl)# policy-age-fileset modThisMonth
bstnA(gbl-fs-age[modThisMonth])# select-files newer-than 1 months
bstnA(gbl-fs-age[modThisMonth])# ...
```

Removing a File Selection

Use the no select-files command to remove a file selection:

```
no select-files {older-than | newer-than}
```

where **older-than | newer-than** chooses the selection to remove.

For example, this removes the "older-than" selection from an age fileset:

```
bstnA(gbl)# policy-age-fileset 2mo
bstnA(gbl-fs-age[2mo])# no select-files older-than
bstnA(gbl-fs-age[2mo])# ...
```

At least one select-files directive is required for the age fileset to function. Without any file-selection criteria, the age fileset selects all files.

Choosing Last-Accessed or Last-Modified

The next step in configuring an age fileset is to determine whether it selects files by last-accessed time or last-modified time. The default is last-modified time. From gbl-fs-age mode, use the last command to choose the age type:

```
last {accessed | modified}
```

where **accessed | modified** is a required choice.

◆ **Note**

The rule that uses this fileset can apply it to files, directories, or both. We do not recommend using last-accessed times for selecting directories in CIFS or multi-protocol (CIFS and NFS) namespaces. CIFS filers update a directory's last-accessed time whenever the policy engine reads the time *stamp; this can cause unexpected directory migrations for directories that have not been accessed by any clients. NFS-only filers do not have this problem.*

For example, the following command set configures the "dayOld" fileset to select its files based on last-accessed time:

```
bstnA(gbl)# policy-age-fileset dayOld
bstnA(gbl-fs-age[dayOld])# last accessed
bstnA(gbl-fs-age[dayOld])# ...
```

Removing the Fileset

Removing a fileset affects file metadata only; it does not delete any files. From gbl mode, use no policy-age-fileset to remove an age fileset:

```
no policy-age-fileset name
```

where ***name*** (1-1024 characters) identifies the fileset to be removed.

You cannot remove a fileset that is referenced by another fileset or used in a rule. The next chapter has configuration instructions for referencing a fileset from a rule.

For example, the following command removes the fileset named "testAge":

```
bstnA(gbl)# no policy-age-fileset testAge
bstnA(gbl)# ...
```

Grouping Files By Offline File Attribute

Another way to define a fileset for CIFS files is by using the CIFS offline file attribute.

Some back-end CIFS servers use hierarchical storage management (HSM) systems to archive file data on a remote server. A small portion of the archived file remains on the CIFS server as a stub, while the bulk of the file is stored remotely. The small, local file has an *offline* attribute set so that when a client, such as the ARX, accesses that file, the HSM restores the full file contents to the CIFS server. If this happens for too many files at one time, such as during a large-scale file migration, the CIFS server's disk could fill up. An attribute-based fileset enables you to identify all files with this CIFS attribute and migrate them (or avoid migrating them) according to a place-rule that specifies how they will be handled.

Creating the Attribute-Based Fileset

Use the policy-cifs-attributes-fileset command to define a fileset based on CIFS attributes, so that migration of the matching files can be controlled.

The command syntax is:

```
policy-cifs-attributes-fileset newfilesetname
```

where *newfilesetname* identifies the fileset that you want to create.

◆ **Note**

This type of fileset is available for namespaces and volumes that support CIFS. It does not apply to an NFS-only namespace, and there is no analogous command in gbl-ns-vol *mode.*

◆ **Note**

The policy functionality only finds the offline file attribute during a scheduled scan of the volume's back-end filers. There are no inline (client-driven) events that indicate that the attribute has changed. Therefore, any place-rule that uses this fileset only assesses the attribute once unless it has a repeating schedule (gbl-ns-vol-plc).

The offline setting is the only CIFS attribute supported in a CIFS-attribute fileset.

Configuring the Offline Attribute Handling

The offline command enables you to specify which setting of the offline attribute the ARX considers meaningful for the purposes of an attribute-based fileset.

The command syntax is:

```
offline {set | not-set | ignore}
```

where set | not-set | ignore is a required choice.

The options behave as follows:

- set indicates that files in this fileset are all offline. In general, F5 does not recommend migrating files that have the offline attribute set. The file's behavior is dependent on the back-end file server and the hierarchical storage manager; they may or may not restore the full file first. Additionally, if the migration causes the file to be restored first, the change in the file may trigger a higher-priority rule and cancel the migration.
- not-set matches only files that are online.
- ignore matches all files, no matter how their offline attribute is set.

Remove the offline configuration by executing the no offline command.

◆ **Note**

The no offline *command is equivalent to* offline ignore.

Example Fileset Creation

For example, to create a fileset named "online" that is based on the offline attribute, type:

```
bstnA(gbl)# policy-cifs-attributes-fileset online
```

This creates a new policy object named "online". Respond to the prompt affirmatively to define the "online" fileset, which can be used by any volume:

```
Create object 'online'? [yes/no] yes
bstnA(gbl-fs-cifs-attr[online])# ...
```

Type the command:

```
bstnA(gbl-fs-cifs-attr[online])# offline not-set
```

to select files for which the offline attribute is off. These files are at full size on the back-end CIFS server on which they reside.

Combining Filesets

You can combine one resulting fileset with others using the commands policy-union-fileset and policy-intersection-fileset. For example, to select all files with the offline attribute unset and in a particular directory, you would:

- specify the particular directory using policy-filename-fileset,
- define a set of online files using the policy cifs-attributes-fileset command and the offline not-set command,

- join those two filesets using the policy-intersection-fileset command, and
- use the intersection fileset in the from (gbl-ns-vol-plc) command of a place-rule.

Removing the Fileset

You can delete an attribute-based fileset by executing this command:

`no policy cifs attributes fileset name`

For example, the following command removes an attribute-based fileset named, “offline_not_set”:

```
bstnA(gbl)# no policy filesize fileset offline_not_set
bstnA(gbl)# ...
```

◆ **Note**

You cannot remove a global fileset if any rule uses it.

Grouping Files by File Name

You can also group files by their names and/or directory paths. From gbl mode, use the policy-filename-fileset command to create a name-based fileset:

policy-filename-fileset *name*

where ***name*** (1-1024 characters) is a required name that you assign to the fileset.

The CLI prompts for confirmation before creating a new fileset; enter **yes** to continue. This puts you into gbl-fs-name mode, where you can optionally specify paths and/or file names for files that belong in the set. By default, a new *filename fileset* matches all files and directories in a volume's root directory, but none below the root.

For example, the following command sequence creates a default named-based fileset, "xmlFiles:"

```
bstnA(gbl)# policy-filename-fileset xmlFiles
This will create a new policy object.

Create object 'xmlFiles'? [yes/no] yes
bstnA(gbl-fs-name[xmlFiles])# ...
```

Setting a Directory Path (optional)

You can set a directory path to narrow the scope of the fileset. Only matching files/subdirectories under this path are included in the fileset; the default is the root directory in the managed volume. From gbl-fs-name mode, use the path exact command to match files in one directory:

path exact *directory* [ignore-case]

where

directory (1-1024 characters) is a directory in any managed volume where the fileset is used, and

ignore-case (optional) matches the above without considering letter case (for example, **path /docs ignore-case** matches "/docs" and "/Docs").

If a volume that uses this fileset does not contain this path, no files match.

For example, the following command set matches files in /www/xml. If this fileset was used in the wwmed~/acct volume, it would match any files in /acct/www/xml:

```
bstnA(gbl)# policy-filename-fileset website
This will create a new policy object.

Create object 'website'? [yes/no] yes
bstnA(gbl-fs-name[website])# path exact /www/xml
bstnA(gbl-fs-name[website])# ...
```

Enabling Recursive Matches (optional)

Recursive matches include files from subdirectories. By default, recursive matches are disabled; you can specify them with a wildcard or regular expression (described below), or you can use the recurse command:

```
recurse
```

For example, the following command set matches files in /www/xml, including all subdirectories:

```
bstnA(gbl)# policy-filename-fileset website
bstnA(gbl-fs-name[website])# path exact /www/xml
bstnA(gbl-fs-name[website])# recurse
bstnA(gbl-fs-name[website])# ...
```

Disabling Recursive Matches

From gbl-fs-name mode, use the no recurse command to disable recursive matches:

```
no recurse
```

For example, the following command set matches files in /log, excluding all subdirectories:

```
bstnA(gbl)# policy-filename-fileset logs
bstnA(gbl-fs-name[logs])# path exact /log
bstnA(gbl-fs-name[logs])# no recurse
bstnA(gbl-fs-name[logs])# ...
```

Matching Against a Wildcard String

To expand your search, you can match against a simple wildcard string to include multiple directories. All matching directories are included in the fileset's scope. To match against wildcards, use the match keyword with a quoted wild-card string:

```
path match "wild-card-string" [ignore-case]
```

where

wild-card-string (1-1024 characters) uses Unix shell conventions, with one exception (below). The quotes are required. Wild-card conventions are summarized as follows:

- ***** is any string of 0 (zero) or more characters, including an empty string.
- **?** is any single character, or no character.

◆ **Note**

*The * and ? match any character, including the "/" character. (The "/" is the Unix delimiter between directories.) Therefore,* ***path match /usr/*/bin*** *matches both "/usr/local/bin" and "/usr/src/mydir/tmp/bin." This may be unexpected for Unix users.*

- **[...]** matches any one of the enclosed characters. For example, [xyz] matches x, y, or z.
- **[a-z]** matches any character in the sorted range, a through z.

- **[^...]** or **[!...]** matches any character that is *not* enclosed. For example, [!xyz] matches any character *except* x, y, or z.

ignore-case (optional) matches the above without considering letter case.

For example, the following command set matches files in any directory named “xml” or “xsl” (upper or lower-case), such as “/www/xml,” “/www/XSL,” “/var/log/Xml,” “/xsl,” or “/XML:”

```
bstnA(gbl)# policy-filename-fileset inXmlOrXslDirs
bstnA(gbl-fs-name[inXmlOrXslDirs])# path match “*/x[ms]l” ignore-case
bstnA(gbl-fs-name[inXmlOrXslDirs])# ...
```

Using a Complex Regular Expression

You can also match directories using a more complex regular expression. For this option, use the regexp keyword along with a quoted regular expression:

```
path regexp “regular-expression” [ignore-case]
```

where

regular-expression (1-1024 characters) uses IBM’s ICU conventions for regular expressions (such as “.*\.htm*”, or “[a-z]*\.txt”). The quotes are required. The section below provides details for the regular-expression syntax.

ignore-case (optional) matches the above without considering letter case.

A regular expression makes it possible to be very specific about your matching criteria.

For example, the following command set uses a regular expression to match all hidden Unix directories (directories that start with “/.” and have something other than “.” as their second character):

```
bstnA(gbl)# policy-filename-fileset hiddenFiles
bstnA(gbl-fs-name[hiddenFiles])# path regexp “/\.[^\.]”
bstnA(gbl-fs-name[hiddenFiles])# ...
```

Regular Expression Syntax

The regular expression syntax follows the ICU regular expression conventions, which allow for multiple patterns in the same expression.

Use the following conventions for simple character matches:

. matches any character (including the “/” delimiter between Unix directories).

***** matches 0 (zero) or more of the preceding character or expression. For example, **.*** matches any string, including the null string.

? matches 0 or one of the preceding character or expression. For example, “z?oo” matches either “zoo” or “aloof.”

+ matches one or more of the preceding character or expression.

\ matches the next character, even if that character is a special character. For example, \. matches a period instead of any character, and \? matches a question mark.

Character Groups

[...] matches any one of the enclosed characters. For example, [xyz] matches x, y, or z.

[a-z] matches any character in the sorted range, a through z.

[^...] matches any character that is *not* enclosed. For example, [^xyz] matches any character *except* x, y, or z.

Shorthand for Character Groups

\d matches any numeric digit, 1-9.

\D matches any character except a numeric digit.

\t matches a <Tab>.

\n, **\f**, and **\r** match various flavors of <Enter>. They are <Newline>, <Form Feed>, and <Carriage Return>, respectively.

\s matches any white-space character, [\t\n\f\r\p{Z}]. The "\p{Z}" is any character with the Unicode property, Z.

\S matches any character that is not white space.

Creating Bounds for the Match

{*a*} matches exactly ***a*** of the preceding character or expression. For example, [xyz]{2} matches exactly two characters where each is x, y, or z.

{*a,b*} matches at least ***a*** but no more than ***b*** of the preceding character or expression. For example, \d{1,4} matches a number that is 1 to 4 digits long.

^... matches the beginning of a line.

...$ matches the end of a line.

Adding Complexity

(...) defines an atom, and

(*atom1*) | (*atom2*) matches both expressions. For example, "(/var)|(/usr)" matches either "/var/log" or "/usr/local/bin."

(*atom1* | *atom2*) also matches both expressions.

atom1* | *atom2 also matches both expressions.

Regular-Expression Samples

^/var matches a path with "/var" at its root (for example, "/var/tmp" or "/variable/data," but not "/bin/var").

^/(var | tmp)/ matches two root directories, "/var/" or "/tmp/".

^/home/[^/]+/$ matches any subdirectory of "/home" (such as "/home/juser/") but does not match any directories below that level (such as "/home/juser/misc/").

For More Information

Consult ICU documentation on the Internet for more details on ICU regular expressions.

Negating the Match

There are some cases where you want to match most paths in the volume, with some exceptions. You can use the not keyword in the path command to negate a match, choosing every path that does *not* match the string:

path exact not *directory* [ignore-case]

matches any directory except the one specified. This only excludes an exact match for *directory*.

path match not *"wild-card-string"* [ignore-case]

matches any directory that does not fit the pattern in the wild-card string.

path regexp not *"regular-expression"* [ignore-case]

matches any directory that does not fit the regular expression.

For example, this command sequence matches all directories in the volume except those named "deleteme:"

```
bstnA(gbl)# policy-filename-fileset keepers
bstnA(gbl-fs-name[keepers])# path match not "*/deleteme/*"
bstnA(gbl-fs-name[keepers])# ...
```

As another example, this command sequence matches all directories except those that start with a "/.":

```
bstnA(gbl)# policy-filename-fileset forAllUsers
bstnA(gbl-fs-name[forAllUsers])# path regexp not "^/\."
bstnA(gbl-fs-name[forAllUsers])# ...
```

Reverting to the Root Path

From gbl-fs-name mode, use the no path command to match files in a volume's root, "/":

no path

For example, the following command set configures a name-based fileset, "acctfiles," to include files in the root:

```
bstnA(gbl)# policy-filename-fileset acctFiles
bstnA(gbl-fs-name[acctFiles])# no path
bstnA(gbl-fs-name[acctfiles])# ...
```

Matching File Names (optional)

You can use the same methods described previously for specifying the fileset's files. These apply to any files in the chosen path(s) from above.

By default, all files match. From gbl-fs-name mode, use the name command to specify certain file names for the fileset:

name exact *file-name* [ignore-case]

name match *"wild-card-string"* [ignore-case]

name regexp *"regular-expression"* [ignore-case]

where

exact *file-name* (1-1024 characters) is an exact file name for the fileset,

match "*wild-card-string*" (1-1024 characters) uses wildcard conventions for shells described above (see *Matching Against a Wildcard String*, on page 13-10),

regexp "*regular-expression*" (1-1024 characters) uses IBM's ICU conventions for regular expressions, also described above (see *Regular Expression Syntax*, on page 13-11), and

ignore-case (optional) matches the above without considering letter case (for example, **name exact deleteme ignore-case** matches both "deleteMe" and "DELETEME").

For example, the following command set matches any file with a ".xml" extension. It searches every directory in the volume's tree, recursively:

```
bstnA(gbl)# policy-filename-fileset xmlFiles
bstnA(gbl-fs-name[xmlFiles])# recurse
bstnA(gbl-fs-name[xmlFiles])# name match "*.xml"
bstnA(gbl-fs-name[xmlFiles])# ...
```

This next fileset uses a regexp to match any file with a ".fm" or a ".pdf" extension:

```
bstnA(gbl)# policy-filename-fileset fm_pdf
bstnA(gbl-fs-name[fm_pdf])# name regexp "\.(fm|pdf)$" ignore-case
bstnA(gbl-fs-name[fm_pdf])# ...
```

Excluding Files

As with paths, you can use the not keyword to select every file that does *not* match the string:

name exact not *file-name* [ignore-case]

matches any file except the one specified. This only excludes an exact match for *file-name*.

name match not *"wild-card-string"* [ignore-case]

excludes any file that fits the pattern in the wild-card string.

name regexp not *"regular-expression"* [ignore-case]

matches any file that does not fit the regular expression.

For example, this command sequence excludes all “*.wmv” and “*.avi” files from the “website” fileset:

```
bstnA(gbl)# policy-filename-fileset website
bstnA(gbl-fs-name[website])# name regexp not “\.(wmv|avi)$”
bstnA(gbl-fs-name[website])# ...
```

Removing the Fileset

Removing a fileset affects file metadata only; it does not delete any files. Use no policy-filename-fileset to remove a name-based fileset:

```
no policy-filename-fileset name
```

where ***name*** (1-1024 characters) identifies the fileset to be removed.

You cannot remove a fileset that is referenced by another fileset or used in a rule. The sections below have configuration instructions for referencing a fileset in these ways.

For example, the following command sequence removes the “httpConf” fileset:

```
bstnA(gbl)# no policy-filename-fileset httpConf
bstnA(gbl)# ...
```

Grouping Files by Size

You can also create filesets based on file size. Each *filesize fileset* contains files "larger-than" or "smaller-than" a size of your choosing, or files in a range between two sizes. From gbl mode, use the policy-filesize-fileset command to create a filesize fileset:

```
policy-filesize-fileset name
```

where ***name*** (1-1024 characters) is the name that you assign to the fileset.

The CLI prompts for confirmation before creating the new fileset; enter **yes** to continue. This puts you into gbl-fs-filesize mode, where you configure the size of files in the fileset.

For example, the following command sequence creates a new filesize fileset:

```
bstnA(gbl)# policy-filesize-fileset veryLarge
This will create a new policy object.

Create object 'veryLarge'? [yes/no] yes
bstnA(gbl-fs-filesize[veryLarge])# ...
```

Selecting Files Based on their Sizes

The next step in configuring a filesize fileset is to determine a size range for its files. You can select files larger-than (or equal-to) a certain size, smaller-than a certain size, or between two sizes. From gbl-fs-filesize mode, use the select-files command to identify the size for the files:

```
select-files {larger-than-or-equal-to | smaller-than} size[k|M|G|T]
```

where

larger-than-or-equal-to | smaller-than is a required choice,

size is any integer, and

k|M|G|T (optional) sets the size units: **k**ilobytes, **M**egabytes, **G**igabytes, or **T**erabytes. The default is bytes if you omit this. All of these values are 2-based, so a kilobyte is 1024 bytes, a megabytes is 1024*1024, and so on. There can be no spaces between the ***size*** and the unit letter; **20M** is correct, but **20 M** is invalid.

You can run this command twice in the same fileset to choose files between two sizes. For example, you can select-files larger-than-or-equal-to 2M and select-files smaller-than 10M to get a set of files between 2 and 10 Megabytes. You cannot run a select-files command that contradicts a previous one; if you already ran select-files larger-than-or-equal-to 2M, you cannot run select-files smaller-than 1k.

For example, these commands create a fileset with a range of sizes:

```
bstnA(gbl)# policy-filesize-fileset veryLarge
bstnA(gbl-fs-filesize[veryLarge])# select-files larger-than-or-equal-to 5G
bstnA(gbl-fs-filesize[veryLarge])# select-files smaller-than 20G
bstnA(gbl-fs-filesize[veryLarge])# ...
```

Removing a File Selection

Use the no select-files command to remove a file selection:

```
no select-files {larger-than-or-equal-to | smaller-than}
```

where **larger-than-or-equal-to | smaller-than** chooses the selection to remove.

For example, this removes the “smaller-than” selection from the above filesize fileset, making it open-ended:

```
bstnA(gbl)# policy-filesize-fileset veryLarge
bstnA(gbl-fs-filesize[veryLarge])# no select-files smaller-than
bstnA(gbl-fs-filesize[veryLarge])# ...
```

Removing the Fileset

Removing a fileset affects file metadata only; it does not delete any files. From gbl mode, use no policy-filesize-fileset to remove a filesize fileset:

```
no policy-filesize-fileset name
```

where ***name*** (1-1024 characters) identifies the fileset to be removed.

You cannot remove a fileset that is referenced by another fileset or used in a rule. The next chapter has configuration instructions for referencing a fileset from a rule.

For example, the following command sequence removes the “testSize” fileset:

```
bstnA(gbl)# no policy-filesize-fileset testSize
bstnA(gbl)# ...
```

Joining Filesets

You can join two or more filesets in a union fileset. A *union* fileset contains all the files in all of its source filesets. A file that is common to two or more of the source filesets is only included once in the resulting union. From gbl mode, use the policy-union-fileset command to create a union fileset:

```
policy-union-fileset name
```

where ***name*** (1-1024 characters) is a required name that you assign to the fileset.

The CLI prompts for confirmation before creating a new fileset; enter **yes** to continue. This puts you into gbl-fs-union mode, where you must identify two or more source filesets to include in the union.

For example, the following command sequence creates an empty union fileset:

```
bstnA(gbl)# policy-union-fileset bulky
This will create a new policy object.

Create object 'bulky'? [yes/no] yes
bstnA(gbl-fs-union[bulky])# ...
```

Identifying the Source Filesets

The final step in configuring a union fileset is to identify two or more source filesets. You can include as many source filesets as desired, but you must have at least two for the union to be meaningful. From gbl-fs-union mode, use the from fileset command to include a source fileset:

```
from fileset fileset-name
```

where ***fileset-name*** (1-64 characters) identifies the source fileset.

For example, the following command set includes three source filesets for the “bulky” union:

```
bstnA(gbl)# policy-union-fileset bulky
bstnA(gbl-fs-union[bulky])# from fileset fm_pdf
bstnA(gbl-fs-union[bulky])# from fileset veryLarge
bstnA(gbl-fs-union[bulky])# from fileset xmlFiles
bstnA(gbl-fs-union[bulky])# ...
```

Removing a Source Fileset

Use the no form of the from fileset command to remove a fileset from the list of sources:

```
no from fileset fileset-name
```

where ***fileset-name*** (1-64 characters) identifies the fileset to remove.

You cannot remove the last source fileset if the union fileset is in use (that is, referenced by another fileset or used in a rule). The next chapter has configuration instructions for referencing a fileset from a rule.

For example, the following command set removes a source fileset from the “bulky” fileset:

```
bstnA(gbl)# policy-union-fileset bulky
bstnA(gbl-fs-union[bulky])# no from fileset xmlFiles
bstnA(gbl-fs-union[bulky])# ...
```

Removing All Source Filesets

Use no from all to remove all source filesets at once:

```
no from all
```

As mentioned above, you cannot remove all source filesets if the union fileset is in use.

For example:

```
bstnA(gbl)# policy-union-fileset testUnion
bstnA(gbl-fs-union[testUnion])# no from all
bstnA(gbl-fs-union[testUnion])# ...
```

Removing the Fileset

Removing a fileset affects the switch configuration only; it does not delete any files. Use no policy-union-fileset to remove a union fileset from the current volume:

```
no policy-union-fileset name
```

where ***name*** (1-1024 characters) identifies the fileset to be removed.

You cannot remove a fileset that is referenced by another fileset or used in a rule. The next chapter has configuration instructions for referencing a fileset from a rule.

For example, the following command sequence removes the “allCustomerSide” fileset:

```
bstnA(gbl)# no policy-union-fileset allCustomerSide
bstnA(gbl)# ...
```

Intersecting Filesets

You can intersect two or more filesets into an intersection fileset. An *intersection* fileset contains files that are common to all of its source filesets; a file must be in all of the source filesets to be included in the intersection. From gbl mode, use the policy-intersection-fileset command to create an intersection fileset:

```
policy-intersection-fileset name
```

where ***name*** (1-1024 characters) is a required name that you assign to the fileset.

As with any fileset, the CLI prompts for confirmation before creating a new intersection fileset; enter **yes** to continue. This puts you into gbl-fs-isect mode, where you identify two or more source filesets to intersect.

For example, the following command sequence creates an empty intersection fileset named “paidBills:”

```
bstnA(gbl)# policy-intersection-fileset paidBills
This will create a new policy object.

Create object 'paidBills'? [yes/no] yes
bstnA(gbl-fs-isect[paidBills])# ...
```

Identifying the Source Filesets

The final step in configuring an intersection fileset is to identify two or more source filesets. You can include as many source filesets as desired, but you must have at least two for the intersection to be meaningful. From gbl-fs-isect mode, use the from fileset command to include a source fileset:

```
from fileset fileset-name
```

where ***fileset-name*** identifies the source fileset.

For example, the following command set intersects two source filesets to make the “paidBills” fileset:

```
bstnA(gbl)# policy-intersection-fileset paidBills
bstnA(gbl-fs-isect[paidBills])# from fileset payments
bstnA(gbl-fs-isect[paidBills])# from fileset customerCorrespondence
bstnA(gbl-fs-isect[paidBills])# ...
```

Removing a Source Fileset

Use the no form of the from fileset command to remove a fileset from the list of sources:

```
no from fileset fileset-name
```

where ***fileset-name*** identifies the fileset to remove.

You cannot remove the last source fileset if the intersection fileset is in use (that is, referenced by another fileset or used in a rule). The next chapter has configuration instructions for referencing a fileset from a rule.

For example, the following command set removes two source filesets from the “paidBills” fileset:

```
bstnA(gbl)# policy-intersection-fileset paidBills
bstnA(gbl-fs-isect[paidBills])# no from fileset underwritten
bstnA(gbl-fs-isect[paidBills])# no from fileset monthlyPaid
bstnA(gbl-fs-isect[paidBills])# ...
```

Removing All Source Filesets

To remove all source filesets with a single command, use no from all:

```
no from all
```

As above, you cannot remove all source filesets if the intersection fileset is in use.

For example, the following command set removes all source filesets from the “paidBills” fileset:

```
bstnA(gbl)# policy-intersection-fileset paidBills
bstnA(gbl-fs-isect[paidBills])# no from all
bstnA(gbl-fs-isect[paidBills])# ...
```

Removing the Fileset

Removing a fileset affects file metadata only; it does not delete any files. Use no policy-intersection-fileset to remove an intersection fileset:

```
no policy-intersection-fileset name
```

where ***name*** (1-1024 characters) identifies the fileset to be removed.

You cannot remove a fileset that is referenced by another fileset or used in a rule. The next chapter has configuration instructions for referencing a fileset from a rule.

For example, the following command sequence removes the “paidBills” fileset:

```
bstnA(gbl)# no policy-intersection-fileset paidBills
bstnA(gbl)# ...
```

Volume-Level Filesets

All of the above filesets are defined globally and can be used in any managed volume. You can also create filesets in specific volumes. Any such fileset can only be used in the volume where you create it. This can be useful for filesets that apply to only a single volume, such as a name-based fileset that references a specific directory path.

You can find all of the above fileset modes under gbl-ns-vol mode, too. These are the commands to create them:

- age-fileset,
- filename-fileset,
- filesize-fileset,
- union-fileset, and
- intersection-fileset.

Each of these commands leads to a mode that is exactly like the modes described above. The only difference is the CLI prompt, which shows that you are under a particular volume. For example, the following command sequence creates a name-based fileset in the “medarcv~/lab_equipment” volume:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# filename-fileset training
bstnA(gbl-ns-vol-fs-name[medarcv~/lab_equipment~training])# path exact /Training
bstnA(gbl-ns-vol-fs-name[medarcv~/lab_equipment~training])# ...
```

Listing all Filesets

Use the show policy filesets command to show all filesets:

`show policy filesets`

This command shows the configuration for all global filesets (configured in gbl mode), followed by all filesets configured in each managed volume.

For example:

```
bstnA(gbl)# show policy filesets

Global Policy:

  Filename Fileset:  website

    Configuration:
      Name Does Not Match Regular Expression:        \.(wmv|avi)$
      Path Is:                                       /www/xml/
      Recurse:                                       Yes
      Rules Using This Fileset:                      0


  Filename Fileset:  hiddenFiles

    Configuration:
      Name Is:
      Path Matches Regular Expression:               /\.[^\.]
      Recurse:                                       No
      Rules Using This Fileset:                      0


  Filename Fileset:  xmlFiles

    Configuration:
      Name Matches Pattern:                          *.xml
      Recurse:                                       Yes
      Rules Using This Fileset:                      0

  Filename Fileset:  fm_pdf

    Configuration:
      Name Matches Regular Expression:               \.(fm|pdf)$ (case ignored)
      Recurse:                                       Yes
      Rules Using This Fileset:                      1


  Filename Fileset:  hiddenFiles

    Configuration:
      Name Is:
      Path Matches Regular Expression:               /\.[^\.]
      Recurse:                                       No
      Rules Using This Fileset:                      0


  File Size Fileset:  veryLarge

    Configuration:
      Rules Using This Fileset:                      1
      Select Files Larger Than:                      5.0 MB
  Fileset Union:    bulky

    Configuration:
      From fileset:                                  fm_pdf
      From fileset:                                  veryLarge
      Rules Using This Fileset:                      1


  Age Fileset:      dayOld

    Configuration:
      Select files older than:                       1 days
      Mode:                                          Last Accessed
      Rules Using This Fileset:                      1


  Filename Fileset:  allDirs

    Configuration:
      Name Is:
      Recurse:                                       Yes
      Rules Using This Fileset:                      2


  Age Fileset:      modEarlier

    Configuration:
      Select files older than:                       1 months
      Mode:                                          Last Modified
      Rules Using This Fileset:                      1


  Age Fileset:      modThisMonth

    Configuration:
      Select files newer than:                       1 months
      Mode:                                          Last Modified
      Rules Using This Fileset:                      1


  Age Fileset:      2mo

    Configuration:
      Select files newer than:                       2 months
      Mode:                                          Last Modified
      Rules Using This Fileset:                      0

Volume:                /vol

Volume:                /acct

Volume:                /rcrds

Volume:                /lab_equipment

Volume:                /test_results

Volume:                /acopia$ns3
```

```
Volume:                /claims

  Filename Fileset:  images

    Configuration:
      Name Is:
      Path Is:                                     /images/
      Recurse:                                     Yes
      Rules Using This Fileset:                    0


Volume:                /acopia$ns4

bstnA(gbl)# ...
```

Showing One Global Fileset

To show a single global fileset, add the global-fileset argument to the end of the show policy filesets command:

show policy filesets global-fileset *fileset-name*

where ***fileset-name*** (optional, 1-1024 characters) chooses the fileset.

For example, the following command shows the “modThisMonth” fileset:

```
bstnA(gbl)# show policy filesets global-fileset modThisMonth

Global Policy:

  Age Fileset:      modThisMonth

    Configuration:
      Select files newer than:                     1 months
      Mode:                                        Last Modified
      Rules Using This Fileset:                    1

bstnA(gbl)# ...
```

Showing Filesets in a Managed Volume

All filesets can be alternatively configured within a managed volume. To show the configuration for such a fileset, specify the namespace and volume name before the fileset name:

show policy filesets namespace *namespace* volume *volume* fileset *fileset-name*

where

namespace (1-30 characters) is the fileset’s namespace.

volume (optional, 1-1024 characters) identifies the fileset’s volume.

fileset-name (optional, 1-1024 characters) chooses the fileset.

Sample - Choosing Files by Name, Location, and Age

The following command sequence creates several filesets as building blocks, then joins them together in union and intersection filesets. This creates a fileset out of *.xls, *.doc, and *.pdf files in /xyz/public that have been accessed in the last 30 days:

First, create a fileset that recursively matches all *.xls files in /xyz/public:

```
bstnA(gbl)# policy-filename-fileset xls_files
This will create a new policy object.

Create object 'xls_files'? [yes/no] yes
bstnA(gbl-fs-name[xls_files])# name match “*.xls”
bstnA(gbl-fs-name[xls_files])# path exact /xyz/public
bstnA(gbl-fs-name[xls_files])# recurse
bstnA(gbl-fs-name[xls_files])# exit
```

Create a similar fileset for *.doc and *.pdf files:

```
bstnA(gbl)# policy-filename-fileset doc_files
This will create a new policy object.

Create object 'doc_files'? [yes/no] yes
bstnA(gbl-fs-name[doc_files])# name regexp “\.(doc|pdf)$”
bstnA(gbl-fs-name[doc_files])# path exact /xyz/public
bstnA(gbl-fs-name[doc_files])# recurse
bstnA(gbl-fs-name[doc_files])# exit
```

Join the two filesets in a union:

```
bstnA(gbl)# policy-union-fileset office_files
This will create a new policy object.

Create object 'office_files'? [yes/no] yes
bstnA(gbl-fs-union[office_files])# from fileset xls_files
bstnA(gbl-fs-union[office_files])# from fileset doc_files
bstnA(gbl-fs-union[office_files])# exit
```

Create an age-based fileset that only takes files modified this month:

```
bstnA(gbl)# policy-age-fileset thisMonth
This will create a new policy object.

Create object 'thisMonth'? [yes/no] yes
bstnA(gbl-fs-age[thisMonth])# select-files newer-than 1 months
bstnA(gbl-fs-age[thisMonth])# end
bstnA#
```

Intersect this latest fileset with the earlier union fileset. The effect is to choose files from a particular directory that were modified this month:

```
bstnA(gbl)# policy-intersection-fileset recent_office_files
This will create a new policy object.

Create object 'recent_office_files'? [yes/no] yes
bstnA(gbl-fs-isect[office_files])# from fileset office_files
bstnA(gbl-fs-isect[office_files])# from fileset thisMonth
bstnA(gbl-fs-isect[office_files])# exit
```

14

Migrating Filesets and Tiering

Overview

You can implement a variety of storage policies, including storage tiering, with placement rules. Each *placement rule* sends a fileset (described in the previous chapter) to one or more shares in the managed volume. You can implement *tiered storage* by migrating various filesets to two or more tiers. For example, files modified in the past week could migrate to a Tier-1 filer and all other files could migrate to a Tier-2 server. This chapter explains how to configure policies for migrating filesets to desired back-end storage.

Direct volumes, which contain no metadata, do not support any filesets. This chapter is relevant to managed volumes only.

Before You Begin

You must configure a namespace and at least one managed volume before you configure any file-placement rules. See *Chapter 7, Configuring a Namespace*, and *Chapter 9, Adding a Managed Volume*.

You must also create one or more filesets for the rule(s) to migrate. See *Chapter 13, Grouping Files Into Filesets*. For regularly-scheduled migrations, you may also want to create a schedule as described in *Chapter 12, Creating a Policy Schedule*.

Concepts and Terminology

When files migrate from one share to another, their parent directories are duplicated on the destination share. The directories are said to be *striped* across the two shares. The original directory, the one that was first-imported, is called the *master directory*. A new file or directory goes to its parent's master directory by default, so that directory trees tend to grow on a single back-end share.

If you have multiple file servers in the same tier, you can group them together as a *share farm*. A share farm can distribute the files among its component shares as directed by your policies; the next chapter describes share farms in detail. You can direct a file migration to a single share or a share farm, as described in this chapter.

Placing Files on Particular Shares

You can use fileset policies to steer files and/or directories onto specific storage targets. You choose the files/directories by configuring a fileset for them, and you choose the storage target by creating a *placement rule* in the volume. The target for file placement can be a storage tier (a share or a share farm) in the current managed volume.

For example, you could direct all of the volume's HTML files to a particular share, or you could migrate an entire directory tree from one share to another. You could also configure the switch to move all files that have not changed for over a week onto a Tier-2 share farm, with slower back-end performance than the servers in Tier 1.

From gbl-ns-vol mode, use place-rule to create a new placement rule:

```
place-rule name
```

where ***name*** (1-1024 characters) is a name you choose for the rule.

The CLI prompts for confirmation before creating the new placement rule; enter **yes** to continue. This puts you into gbl-ns-vol-plc mode, where you must choose a source fileset, set the storage target for the files, and enable the rule. There are also some optional commands you can invoke from this mode.

For example, the following command sequence creates an empty placement rule, "docs2das8," for the volume, "wwmed~/acct:"

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8

This will create a new policy object.

Create object 'docs2das8'? [yes/no]   yes
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

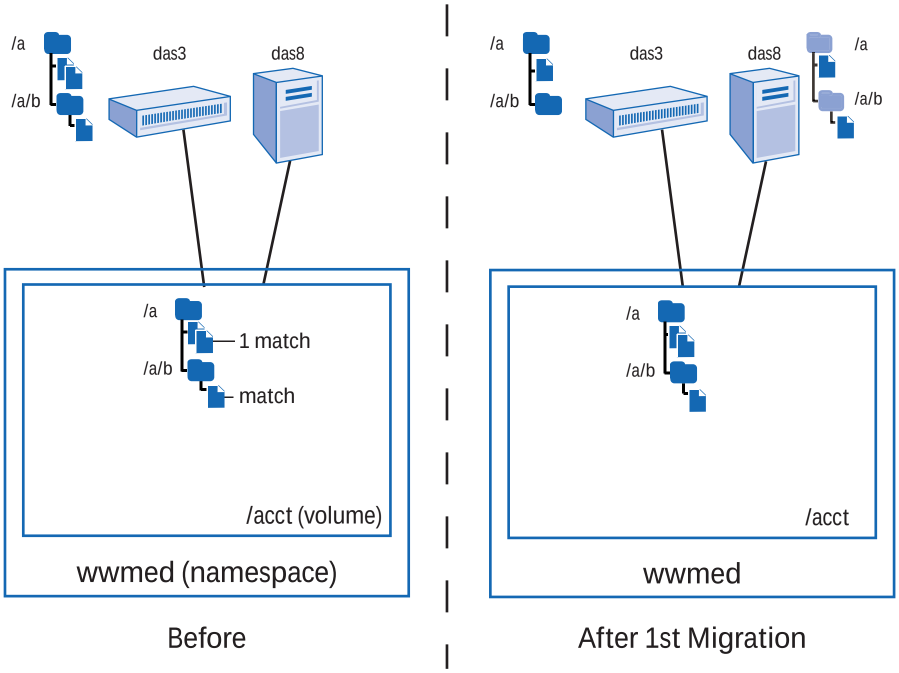
As another example, this command sequence creates an empty placement rule, "busy2tier1," for the volume, "medarcv~/lab_equipment:"

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule busy2tier1

This will create a new policy object.

Create object 'busy2tier1'? [yes/no]   yes
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# ...
```

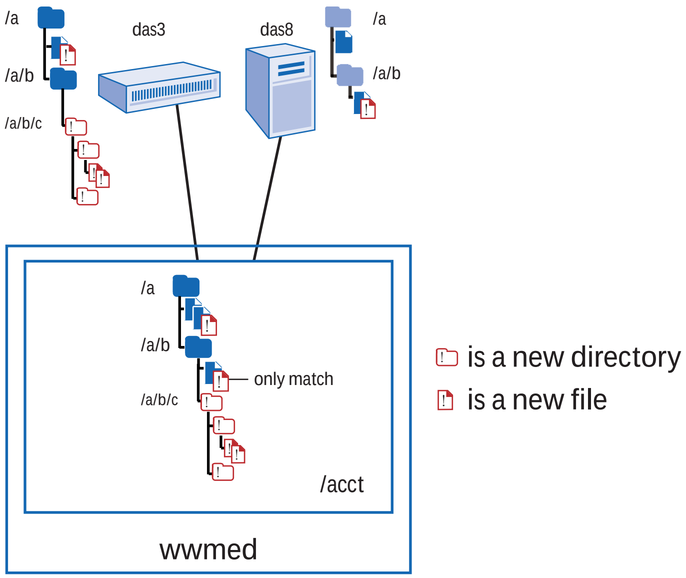
Identifying the Source Fileset

The next step in configuring a placement rule is to identify the source fileset. This chooses a set of files and/or directories based on their names, sizes, ages, or other criteria; this set of files and/or directories changes as clients create, edit, and delete them in the volume. The placement rule migrates all selections from the source fileset onto the target storage, then it steers any new matching files/directories as clients create and edit them.

You can make the placement rule select files only, directories only, or both; by default, the rule selects files only. The default applies the fileset exclusively to file names. Directories are copied (or *striped*) only as needed to hold the placed files. For example, the illustration below shows a small directory tree on das3 containing two matching files. The directories that contain the matching files, /a and /a/b, are striped to das8 so that das8 can hold those files. One file in directory /a does not match, so it remains in the master directory on das3.



Note that the master copies of the directories remain on their original filer, das3; this means that new files in those directories go to das3 by default, if they are outside the fileset. All of their new subdirectories go to das3, too. In this illustration, only one new file matches and is steered onto das8. All new directories and non-matching files are created on das3, which has all of the master directories. Had the rule been removed after the initial migration, above, *all* of the new files would have followed their master directories onto das3.



From gbl-ns-vol-plc mode, use the from fileset ... match files command to apply a source fileset to files only. This results in the behavior illustrated above:

```
from fileset fileset-name [match files]
```

where

fileset-name (1-1024 characters) identifies the source fileset, and

match files (optional) applies the fileset only to files, not directories. This is the default, so it is optional.

For example, the following command set selects files that match the “fm_pdf” fileset:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# from fileset fm_pdf match files
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

These commands select files that match the “modThisMonth” fileset, an age-based fileset from examples in the previous chapter:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule busy2tier1
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# from fileset modThisMonth
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# ...
```

Filesets and Hard Link Migration

To migrate hard links off of the share, you need to use the source command without any fileset to drain all of the files and directories off of the share, or you can use the migrate hard-links command to enable migration of files that match a fileset and have hard links.

Hard link migration is disabled by default, and is configured as no migrate hard-links.

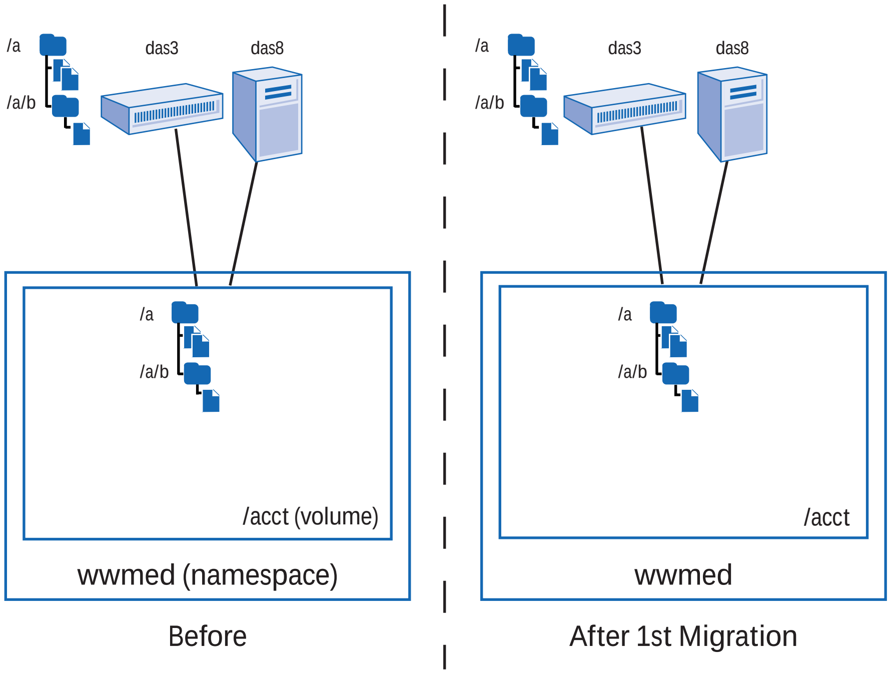
The migrate hard-links command requires the place rule to be configured to use a source share, and that share cannot be used by another place rule. When the migrate hard-links command is executed and the place rule does not have a source share configured, an error message is displayed.

This feature is for NFS services only.

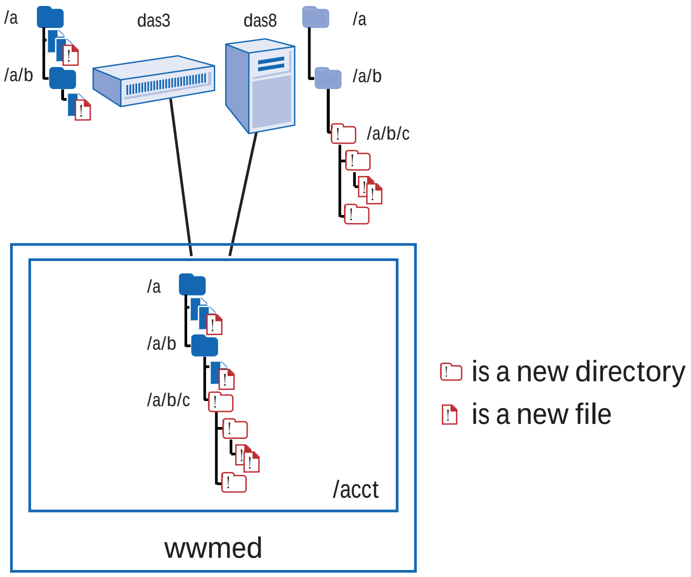
Matching Directories Only

Consider a directory tree on a filer that is filled to a comfortable level, but clients are likely to add subdirectories that may overfill the share. You want all new subdirectories to be created on a new filer and grow there, but existing files and directories can stay. Also, new files in the existing directories can stay on the original filer.

You can use the from fileset command to select directories only, so that the fileset's criteria is not applied to any files. This steers new directories to das8. Old directories keep their masters on das3, but new directories are created at das8 and therefore have their masters there. Files are not matched, so none of the existing files migrate. The initial run of the rule does nothing:



This configuration mainly focuses on new directories and their sub-trees. The placement rule steers new directories to das8. Since the new directories are created on das8, the das8 instance of any new directory is master. By default, all of its child files and directories follow it onto das8. Directory /a/b/c therefore grows on das8, as would any other new directories in the volume. Files are not matched by this rule, so they always go to the filer that holds their parent's master directory; the old directories (/a and /a/b) therefore store all new files on das3, and files in the new directories go to das8. This configuration is designed to move all of the major tree growth onto the target filer.



In the from fileset command, you can use the match directories argument to choose directories only. This causes the placement illustrated above:

```
from fileset fileset-name match directories
```

where

fileset-name (1-64 characters) identifies the source fileset, and

match directories applies the fileset only to directories, not files.

For example, the following command set makes an all-inclusive fileset and uses it to match all directories:

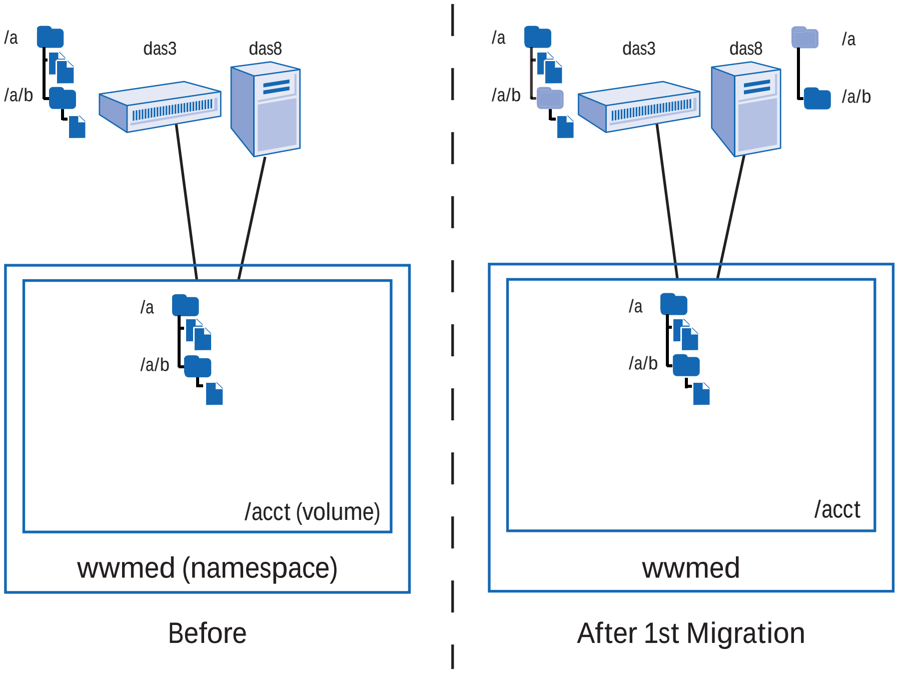
```
bstnA(gbl)# policy-filename-fileset all
bstnA(gbl-ns-vol-fs-name[all])# recurse
bstnA(gbl-ns-vol-fs-name[all])# exit
bstnA(gbl)# namespace wwmed
bstnA(gbl-ns[wwmed])# volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule noNewDirs
bstnA(gbl-ns-vol-plc[wwmed~/acct~noNewDirs])# from fileset all match directories
bstnA(gbl-ns-vol-plc[wwmed~/acct~noNewDirs])# ...
```

Matching and Promoting Directories

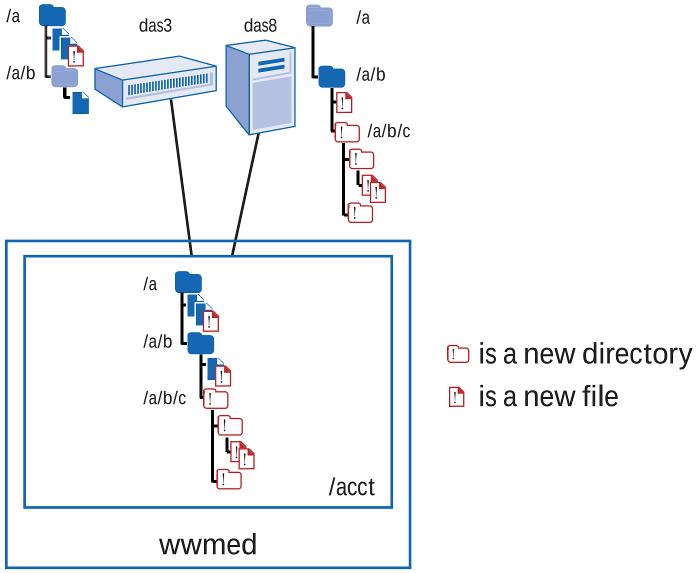
Both of the previous from fileset commands did not change the current master directories; das3 kept the master versions of both /a and /a/b. You may want to promote the migrated directories to master, so that the source filer keeps all existing files but the target filer gets all of their *new* files and subdirectories by default. This can be useful after rebuilding a managed volume with several filers: a directory tree that was previously constrained to das8 may be re-imported with its masters on other filers. By promoting all matching directories to master, you can return sole mastership to das8. This causes the directories to grow on das8, not on various filers throughout the volume.

(For instructions on rebuilding a volume, refer to *Rebuilding a Volume*, on page 7-33 of the *ARX CLI Maintenance Guide*.)

For example, suppose das8 is supposed to be master of /a/b. You can promote it and all of its descendant directories without migrating any files. After the placement rule runs, the file already under /a/b stays on das3. The master for /a/b is now on das8. The /a directory is striped to das8 so that it can hold the /a/b directory; its master remains on das3.



As clients add new files and subdirectories to /a/b, they go onto das8 instead of das3. New files in /a, which is outside the fileset, continue to gravitate to das3.



In the from fileset command, you can add the promote-directories flag to promote the chosen directories. This causes the placement illustrated above:

```
from fileset fileset-name match directories promote-directories
```

where

fileset-name (1-64 characters) identifies the source fileset,

match directories applies the fileset only to directories, not files, and

promote-directories promotes all matching directories on the target filer to master.

For example, the following command set creates a fileset to match the /a/b tree, matches the fileset against directories, and promotes the matching directories:

```
bstnA(gbl)# policy-filename-fileset a_b_tree
bstnA(gbl-ns-vol-fs-name[a_b_tree])# path match /a/b
bstnA(gbl-ns-vol-fs-name[a_b_tree])# recurse
bstnA(gbl-ns-vol-fs-name[a_b_tree])# exit
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule reset
bstnA(gbl-ns-vol-plc[wwmed~/acct~reset])# from fileset a_b_tree match directories
promote-directories
bstnA(gbl-ns-vol-plc[wwmed~/acct~reset])# ...
```

Best Practice for Tiering: Promote all Directories In Tier 1

If the back-end filer with a master directory goes offline, the managed volume’s clients can no-longer access the directory or its contents. Tier 1 storage is, by definition, more reliable than its lower-tier counterparts, so we recommend storing all master directories on that tier. You can use a filename fileset to select all directories in the volume, then use a placement rule command to promote all of their Tier-1 instances to master.

For example, the following command sequence creates an all-inclusive fileset and uses it to place all master directories onto a “teir1” share farm:

```
bstnA(gbl)# policy-filename-fileset allDirs

This will create a new policy object.

Create object 'allDirs'? [yes/no] yes
bstnA(gbl-fs-name[allDirs])# recurse
bstnA(gbl-fs-name[allDirs])# exit
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule masterDirs2Tier1

This will create a new policy object.

Create object 'masterDirs2Tier1'? [yes/no] yes
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~masterDirs2Tier1])# from fileset allDirs match
directories promote-directories
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~masterDirs2Tier1])# target share-farm tier1
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~masterDirs2Tier1])# enable
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~masterDirs2Tier1])# exit
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# ...
```

We recommend that you use a rule like this in conjunction with a high import priority for all of your Tier-1 shares. Recall *Setting the Share’s Priority (for Tiering)*, on page 9-36. This ensures that directory masters are on Tier 1 shares from the moment that the volume imports them.

Promoting Directories on a Target Share Farm

If the file-placement target is a share farm, as above, the share that gets the directory also gets the directory promotion. The share farm’s directory constraints or new-file-placement algorithm determines the exact share where the directory goes. The next chapter describes share farms in detail, as well as methods for constraining and/or balancing files in a share farm. You can use the instructions in that chapter to determine which directories go to which shares in the farm.

Avoid Promoting CIFS Directories Based on Last-Accessed Time

CIFS filers update a directory’s last-accessed time whenever the policy engine reads the time stamp. Do not migrate and promote CIFS directories if the source fileset

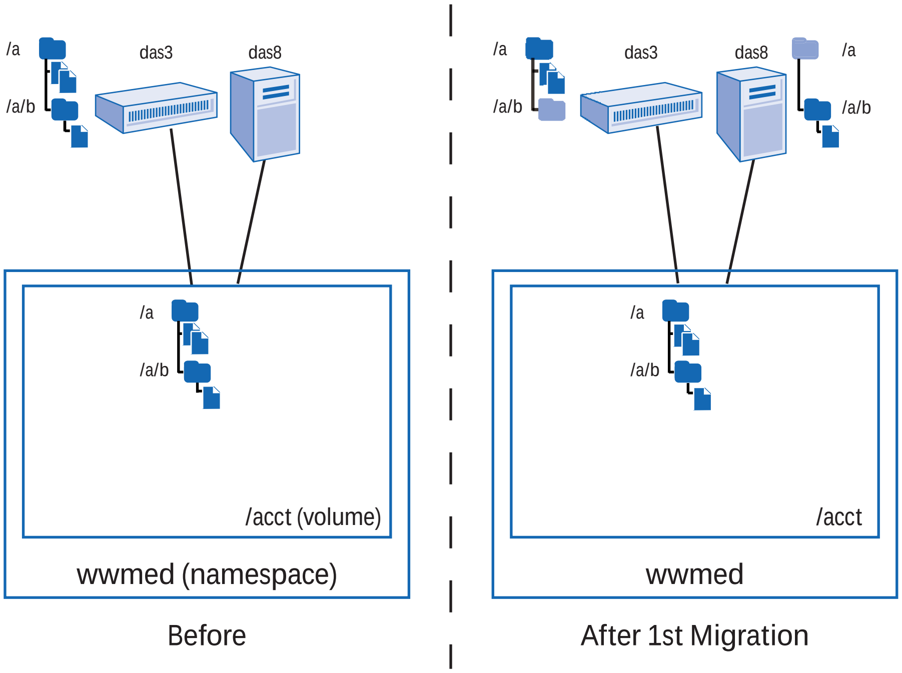
- is age-based, and
- does all selections based in the last-accessed time (recall *Choosing Last-Accessed or Last-Modified*, on page 13-4).

If a rule migrates and promotes directories based on their last-accessed times in a CIFS or multi-protocol namespace, the back-end filer changes the directory's time stamp after the migration. Any other rule that chooses directories based on the last-accessed time would therefore use the wrong time stamp. This can cause unpredictable results.

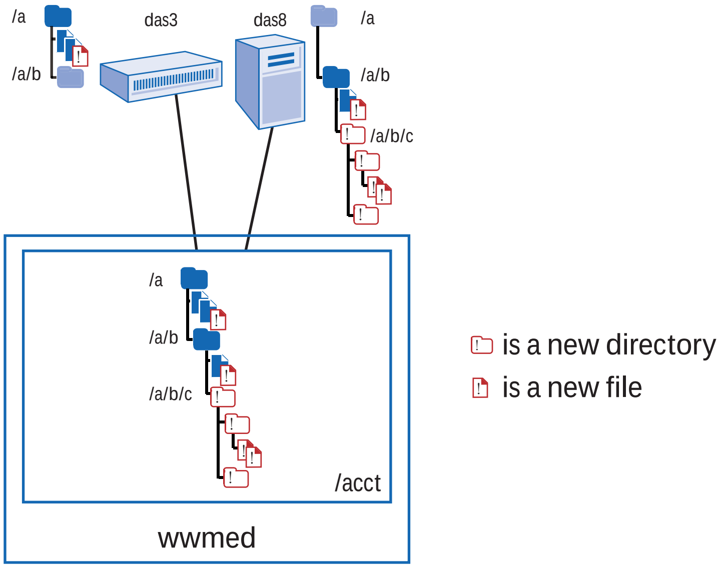
NFS-only filers do not have this problem, nor is it a problem with files.

Matching Directory Trees (Directories and Files)

By combining files, directories, and directory promotion, you can move an entire directory tree from das3 to das8 and make it grow on das8. For example, you can migrate all existing files in directory /a/b in addition to ensuring that new files and directories get created on das8. The only difference between this and the previous example is that the existing file in /a/b also migrates over to das8:



Each new file and directory, as in the previous example, follows its parent's master directory. Directory /a/b grows on das8 while new files in /a continue to gravitate to das3:



In the from fileset command, you can use match all to match both files and directories. Keep the promote-directories flag, to make the selected directories (such as /a/b/c) "master" at the target filer. This causes the placement illustrated above:

```
from fileset fileset-name match all promote-directories
```

where

fileset-name (1-64 characters) identifies the source fileset,

match all applies the fileset to both files and directories, and

promote-directories promotes all matching directories on the target filer to master.

For example, the following command set migrates the directories that match the "a_b_tree" fileset:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule mvtree
bstnA(gbl-ns-vol-plc[wwmed~/acct~mvtree])# from fileset a_b_tree match all promote-directories
bstnA(gbl-ns-vol-plc[wwmed~/acct~mvtree])# ...
```

Choosing the Target Storage

The target storage for the fileset is a share or share farm in the current managed volume. The file-placement rule sends all matching files and/or directories to the share(s) you choose here. From gbl-ns-vol-plc mode, use one of two target rules to set the fileset's storage target:

target share *share-name*

where ***share-name*** (1-64 characters) is a share from the current volume. Use the show global-config namespace command to see the shares in each volume: see *Showing Namespace Configuration*, on page 7-26.

target share-farm *share-farm-name*

where ***share-farm-name*** (1-1024 characters) is a share farm within the current volume. The show global-config namespace command also shows the share farms in each of the namespace's volumes.

For example, the following command sequence selects a share, "bills," as the home for all files matched by the "docs2das8" rule. (The volume share, "bills," maps to a share on the "das8" filer.)

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# target share bills
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

As another example, the following command sequence selects a share farm, "tier1," as the target for a different placement rule:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule busy2tier1
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# target share-farm tier1
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# ...
```

Automatically Closing All Open Files (CIFS)

This section is only relevant in a volume that supports CIFS. Skip this section if your file-placement rule is in an NFS-only volume.

To safely migrate a file between shares, the policy engine must open it without granting write access to any CIFS clients. This is impossible if a CIFS-client application already has the file open for a write or delete operation. The rule reacts to such a file by periodically retrying until the client closes the file or the migration is otherwise finished, whichever comes first. If any file is persistently open while the placement rule runs, the file remains stranded on the source share.

You can authorize the file-placement rule to close any open file and keep it closed until the migration finishes. From gbl-ns-vol-plc mode, use the migrate close-file command to allow the rule to automatically close any open files it encounters:

```
migrate close-file
```

◆ **Note**

This does not close any files opened through NFS. It is therefore possible for an NFS client to prevent a file migration in a multi-protocol (CIFS and NFS) volume.

For example, the following command sequence permits the "dailyArchive" rule to close any open files:

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# place-rule dailyArchive
bstnA(gbl-ns-vol-plc[medarcv~/rcrds~dailyArchive])# migrate close-file
bstnA(gbl-ns-vol-plc[medarcv~/rcrds~dailyArchive])# ...
```

Excluding a Fileset

A CIFS client gets a sharing-violation error if he or she tries to access a file that the volume is holding closed. Some files in the source fileset may be too important to be held closed, even for a short time. You can select such files with another fileset, and then use that fileset in the optional exclude clause:

```
migrate close-file exclude fileset-name
```

fileset-name (optional, 1-64 characters) identifies the fileset to exclude from the auto-close feature.

CIFS clients can open any file in the excluded fileset during the migration. If the client holds the file open for the duration of the rule run, the rule cannot migrate the file.

For example, the following command sequence excludes a fileset, "CEO_files," from auto-closure by the "dailyArchive" rule:

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# place-rule dailyArchive
bstnA(gbl-ns-vol-plc[medarcv~/rcrds~dailyArchive])# migrate close-file exclude CEO_files
bstnA(gbl-ns-vol-plc[medarcv~/rcrds~dailyArchive])# ...
```

Showing Auto-Closed Files

You can use the show policy files-closed command to show all files that have been auto-closed by a particular managed volume:

```
show policy files-closed namespace namespace volume volume [rule
rule-name]
```

where

namespace (1-30 characters) is the namespace,

volume (1-1024 characters) identifies the managed volume, and

rule-name (optional, 1-1024 characters) identifies a specific placement rule in the volume.

This shows a separate table for each chosen rule. The table contains one row per auto-closed file, with the time the file was closed and the path to the file. The time is in UTC, not local time. The path is a virtual path, starting at the root of the managed volume.

For example, the following command sequence shows one file has been auto-closed by the “dailyArchive” rule:

```
bstnA(gbl)# show policy files-closed namespace medarcv volume /rcrds

Namespace:        medarcv
Volume:           /rcrds
Rule
----------------------------------------
dailyArchive

Time                  Filename
--------------------  ------------------------------------------------------------
2007-10-01T07:28:58   /recoveryStats/examSchedule.doc
Rule
----------------------------------------
masterDirs2Rx

Time                  Filename
--------------------  ------------------------------------------------------------
bstnA(gbl)# ...
```

Manually Closing Open Files

You can opt to manually monitor the open files in the rule’s path and selectively close them. Methods for using this manual approach are summarized below. From gbl-ns-vol-plc mode, use no migrate close-file to revert to a manual method of finding and closing open files:

```
no migrate close-file
```

For example, this command sequence chooses a manual method for dealing with open files in the path of the “emptyRH” rule:

```
bstnA(gbl)# namespace insur volume /claims
bstnA(gbl-ns-vol[insur~/claims])# place-rule sprdsheet_mig
bstnA(gbl-ns-vol-plc[insur~/claims~sprdsheet_mig])# no migrate close-file
bstnA(gbl-ns-vol-plc[insur~/claims~sprdsheet_mig])# ...
```

This setting creates an issue for the placement rule. If any client holds a CIFS file open during the file migration, the rule retries the migration, and the migration may eventually fail for that file. If a CIFS-client application holds a file open for the duration of the rule’s run, the migration cannot succeed for that file.

You can use the show cifs-service open-files command to get a list of all open, read-locked files (see *Listing Open Files in a CIFS Service*, on page 11-15 of the *ARX CLI Maintenance Guide*). To close one from the CLI, use close cifs file (see *Closing an Open File*, on page 11-18 of the same manual).

Limiting Each Migration (optional)

You can use the limit-migrate command to put a ceiling on file-migration throughput. The policy engine migrates files until it meets this limit; it stops migrating as soon as it discovers that the next file would exceed the limit.

```
limit-migrate size[k|m|g|t]
```

where

size (1-18,446,744,073,709,551,615) is the size, and

k|m|g|t (optional) is the units; **k**ilobytes (1024 bytes), **m**egabytes (1024*1024 bytes), **g**igabytes (1024*1024*1024), or **t**erabytes (1024*1024*1024*1024). The default is bytes.

This limit applies to every run of the placement rule, so you can use it in conjunction with a schedule to migrate a limited amount of data during off-hours. For example, you can allow a limit of 10 Gigabytes and a daily schedule that runs at midnight. If the source fileset contains 50 Gigabytes of data, it would migrate over five nights, 10G per night.

This limit has no effect on a placement rule that matches directories only. The from fileset command determines whether files, directories, or both are matched. This command is described above.

For a placement rule without a schedule, this limit applies to the one-and-only run of the rule. If the original fileset exceeds this limit, the left-over files from that fileset remain on their source share(s) indefinitely. New files that belong in the fileset are created at the target share(s), before they have any size, so they are not blocked by this limit.

For example, the following command sequence sets a 50G limit on the “docs2das8” rule:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# limit-migrate 50g
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

Removing the Limit

By default, a placement rule migrates until all matching files are removed from the source share(s). Use the no limit-migrate command to return to this default:

```
no limit-migrate
```

For example, the following command sequence removes any migration limit in the “nonbusy2tier2” rule:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule nonbusy2tier2
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~nonbusy2tier2])# no limit-migrate
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~nonbusy2tier2])# ...
```

Using Staged Migrations (optional)

If your placement rules tend to move very large files, you may wish to migrate files with a staged method. A staged migration transfers a file to a hidden staging directory on the target share, which can take a long time for large files, and then moves the file to the intended target directory after the network transfer is finished. The final move is instant. This avoids a problem with file migrations that require long transfer times; if a file migrates directly to its target directory and a snapshot occurs during the migration, the snapshot operation shuts down the migration. The migration must restart from the beginning the next time the rule runs. Staged migrations are insulated from snapshots.

From gbl-ns-vol mode, you can use the policy migrate-method staged command to start using staged migrations in the current volume:

```
policy migrate-method staged
```

For example, the following command sequence sets up staged migrations the “wwmed~/acct” volume:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# policy migrate-method staged
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Reverting to Direct Migrations

Direct file migrations are slightly faster than staged migrations in a volume that migrates large numbers of small files. However, they can be interrupted by snapshot operations, and this can be a problem for a volume with very-large files.

Direct migrations are the default. Use the no policy migrate-method command to return to direct migrations:

```
no policy migrate-method
```

For example, the following command sequence restores the direct-migration method to the “medarcv~/lab_equipment” volume:

```
bstnA(gbl)# namespace ns1 volume /vol
bstnA(gbl-ns-vol[ns1~/vol])# no policy migrate-method
bstnA(gbl-ns-vol[ns1~/vol])# ...
```

Disabling Inline Notifications (optional)

Clients make changes to files that may cause them to be selected by a file-placement rule; for example, a client could rename a file or change its size. By default, the file-placement rule monitors all client changes *inline* and migrates any files that newly-match the source fileset. This occurs on an unscheduled basis. You have the option to wait for the next scheduled volume scan; from gbl-ns-vol-plc mode, use no inline notify to disable the rule’s inline notifications:

```
no inline notify
```

This has no effect on *new* files or directories; the rule directs these so that they are created on the target share. No migrations occur in this case.

We recommend that you run this command only in a file-placement rule with a schedule (described in the next section). Unless the rule runs on a schedule, this command disables all migrations after the rule's initial run.

If this rule is part of a tiered-storage solution, where files migrate from better filers to lesser ones as they get older, you should disable inline notifications for non tier-1 rules only. Tier 1 rules should keep inline notify enabled. That is, the file-placement rule for tier 1 should migrate files to tier 1 as soon as a client changes the file, but migrations to other tiers should wait until the next scheduled run (with no inline notify). This configuration reduces the computation load on the ARX.

For example, the following command sequence turns off inline notifications to the "nonbusy2tier2" rule:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule nonbusy2tier2
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~nonbusy2tier2])# no inline notify
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~nonbusy2tier2])# ...
```

Re-Enabling Inline Notifications

Inline notifications are recommended for rules without schedules, or for tier-1 rules in a tiered-storage solution. To reinstate inline notifications, use the inline notify command:

```
inline notify
```

For example, the following command sequence reinstates inline notifications for the "busy2tier1" rule:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule busy2tier1
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# inline notify
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# ...
```

Applying a Schedule

You assign a regular schedule to a placement rule that uses an age-based fileset, such as a tiered-storage rule. Every time the schedule fires, the rule assesses the age of all files based on the current time. For example, consider a fileset of files newer-than 1 week. If the schedule fires on 1/12, it takes all files modified between 1/6 and today (1/12); if the schedule fires again on 1/13, it takes all files modified between 1/7 and today (now 1/13). Each time the schedule fires, the age-based fileset finds a new group of files based on that time.

Without a schedule, the placement rule only fires once, when it is enabled, assesses and migrates its files then, and never includes any other files as they age. The time is never re-assessed, so the files in the age-based fileset can never change: it only includes files modified between 1/6 and 1/12, forever.

Chapter 12, Creating a Policy Schedule has full details on creating a schedule. To apply a schedule to the placement rule, use the schedule command in gbl-ns-vol-plc mode:

```
schedule name
```

where ***name*** (1-64 characters) identifies the schedule. Use the show schedule command to list all schedules.

For example, the following command sequence applies a daily schedule to the “nonbusy2tier2” rule:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule nonbusy2tier2
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~nonbusy2tier2])# schedule daily4am
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~nonbusy2tier2])# ...
```

Optional for Non-Age-Based Filesets

For placement rules whose filesets are *not* based on age (such as file-name and size filesets), a schedule is optional. These placement rules migrate all matching files as soon as you enable them. New files are directed to the proper back-end share if they match. By default, files are also migrated as soon as a client changes them (renames them or changes their size) so that they newly-match the fileset. For these reasons, a schedule is generally unnecessary for these filesets.

However, there are situations where a schedule is useful for these filesets, too:

- The fileset is very large, so you want to run the migration in stages, possibly during off-peak hours. To spread the load out over time, you can use duration in the schedule or limit-migrate in the placement rule. A schedule is required to periodically resume the migration.
- You disabled *inline* notifications and migrations, as described above. An inline migration occurs when a client renames a file or changes its size so that it newly matches a placement rule’s fileset. With inline migrations disabled, a schedule would migrate all of the newly-matching files on a

regular basis, rather than the moment when they each match the fileset. Inline notifications are typically disabled for Tier-2 or Tier-3 placement rules.

For example, the following command sequence applies a daily schedule to the “mvTars” rule:

```
bstnA(gbl)# namespace archives volume /home
bstnA(gbl-ns-vol[archives~/home])# place-rule mvTars
bstnA(gbl-ns-vol-plc[archives~/home~mvTars])# schedule daily4am
bstnA(gbl-ns-vol-plc[archives~/home~mvTars])# ...
```

Rule-Configuration Changes in a Scheduled Rule

After you apply a schedule to an enabled rule, other configuration changes wait until the next time the schedule fires. For example, if you run no inline notify on a file-placement rule with a schedule, the rule continues to process inline changes until the next scheduled run. This only applies to an enabled rule, one that is already running; the method for enabling a rule is described later in the chapter.

You can make a rule change effective immediately by disabling and then re-enabling the rule.

One Volume Scan for File-Placement Rules on the Same Schedule

A file-placement rule performs a *volume scan* every time it runs, so multiple file-placement rules on the same schedule share a single volume scan. That is, the ARX scans the volume’s back-end shares once and then all file-placement rules use the same scan results.

This only applies to file-placement rules with the same set of source shares (you can select specific source shares in order to drain them, as described in *Identifying the Source Share(s)*, on page 10-12 of the *ARX CLI Maintenance Guide*). If two or more placement rules are migrating off of the same source share(s) and using the same schedule, the ARX scans the share(s) only once.

Removing the Schedule

Use no schedule to use only a single, initial volume scan and migrate all files and/or directories found in that scan. This makes the placement rule migrate all existing files and/or directories in a single session:

```
no schedule
```

Again, a schedule (or lack of schedule) has no effect on new files, or files that clients change to match the fileset.

For example:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# no schedule
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

Configuring Progress Reports

The next step in configuring a placement rule is optional but strongly recommended: setting up progress reports. Progress reports show all the milestones and results of a file-placement execution. The policy engine generates a report each time the schedule fires and invokes the rule. If the rule has no schedule, the rule generates a single report when it first runs.

By default, a placement rule generates no reports. From gbl-ns-vol-plc mode, use the report command to generate reports for the current rule:

```
report prefix
```

where ***prefix*** (1-1024 characters) is the prefix to be used for the rule's reports. Each report has a unique name in the following format: *prefixYearMonthDayHourMinute*.rpt (for example, xrayBkup200903031200.rpt for a report with the "xrayBkup" prefix).

This only generates a report if the rule executes some scheduled action, such as scanning a back-end filer or migrating at least one file or directory. It does not report on placement of new files, created by clients through the VIP, or on migrations that occur inline.

For example, the following command sequence enables reporting for the "dos2das8" rule:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# report docsPlc
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

Generating Verbose Reports

Placement reports are terse by default. To make them verbose, use the optional verbose flag at the end of the report command:

```
report prefix verbose
```

where ***prefix*** is explained above.

For example, the following command resets "docs2das8" to produce verbose reports:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# report docsPlc verbose
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

Deleting Empty Reports

By default, a placement rule creates a report every time it executes some action (such as scanning a back-end filer), even in cases where no files or directories are migrated. This results in empty reports. To prevent these empty reports, use the optional delete-empty flag at the end of the report command:

```
report prefix [verbose] delete-empty
```

where ***prefix*** and **[verbose]** are explained above.

For example, the following command sets “busy2tier1” to delete any empty reports:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule busy2tier1
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# report docsPlc verbose delete-empty
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# ...
```

Keeping Only Reports Showing Errors

You can also choose to keep only the reports that show errors. Use the error-only flag to make the rule delete any report unless it contains at least one error:

report *prefix* [verbose] error-only

where ***prefix*** and **[verbose]** are explained above.

For example:

```
bstnA(gbl)# namespace swic volume /users
bstnA(gbl-ns-vol[swic~/users])# place-rule active_to_new_srvrs
bstnA(gbl-ns-vol-plc[swic~/users~active_to_new_srvrs])# report daily active_files_2_t1 error-only
bstnA(gbl-ns-vol-plc[swic~/users~active_to_new_srvrs])# ...
```

Finding and Showing File-Placement Reports

You can use the show reports type Plc command to list all file-placement reports on the ARX. You can invoke this command from any mode:

show reports type Plc

Each report name contains a time stamp to help you identify the report you want. To show the report’s contents, use show reports *report-name* from any mode:

show reports *report-name*

For example, the following command sequence lists several file-placement reports and then shows one of them from the “docsPlc” rule:

```
bstnA(gbl)# show reports type Plc


  reports
    Codes: Plc=Place Rule
      daily_archive_201003120637.rpt Mar 12 06:37  2.3 kB        Plc  DONE: 0 in 00:00:10
      daily_archive_201003120717.rpt Mar 12 07:17  2.3 kB        Plc  DONE: 0 in 00:00:50
      daily_archive_201003120734.rpt Mar 12 07:34  2.3 kB        Plc  DONE: 0 in 00:00:11
      docsPlc_20100312063557.rpt Mar 12 06:36  1.8 kB        Plc  DONE: 0 in 00:00:35
      docsPlc_20100312063558.rpt Mar 12 06:36  16 kB        Plc  DONE: 68 in 00:00:33
      docsPlc_20100312071709.rpt Mar 12 07:17  1.8 kB        Plc  DONE: 0 in 00:00:10
      drain_share_rule_for_share_it5_201003120708.rpt Mar 12 07:08  2.3 kB        Plc  DONE: 44 in 
00:00:11
      mvDats_201003120731.rpt Mar 12 07:32  2.3 kB        Plc  DONE: 0 in 00:00:10

bstnA(gbl)# show reports docsPlc_20100312063558.rpt
**** File Placement Report: Started at Fri Mar 12 06:35:58 2010 ****
**** Software Version: 5.02.000.12545 (Mar  9 2010 20:14:21) [nbuilds]
**** Hardware Platform: ARX-4000
**** Report Destination:

  Place Rule:       docs2das8
```

```
    Configuration:
      Namespace:                                     wwmed
      Volume Name:                                   /acct
      From fileset:                                  bulky (files only)
      Target share:                                  bills
      Report:                                        docsPlc
      Report Verbose:                                Enabled
      Report Delete Empty:                           Disabled
      Report Errors Only:                            Disabled
      Inline Report:                                 docsPlc
      Inline Report Interval:                        hourly
      Inline Report Verbose:                         Enabled
      Inline Report Delete Empty:                    Disabled
      Inline Report Errors Only:                     Disabled
      Migrate limit:                                 50G
      Volume Scan:                                   Enabled
      Inline Notifications:                          Enabled
      Promote Directories:                           Disabled
      Auto-Close Files:                              Disabled

      Tentative:                                     No
      State:                                         Enabled


  Date                       Source Share     Target Share
      File
      Result
  --------------------------------------------------------------------------
  Fri Mar 12 06:36:09 2010   it5              bills
      /acct/shellCmds_winXP.fm
      Complete
  Fri Mar 12 06:36:09 2010   it5              bills
      /acct/intro.fm
      Complete
  Fri Mar 12 06:36:09 2010   it5              bills
      /acct/usersGroups.fm
      Complete
  Fri Mar 12 06:36:10 2010   it5              bills
      /acct/connectathon.fm
      Complete
  Fri Mar 12 06:36:10 2010   it5              bills
      /acct/rework_vpn.fm
      Complete
  Fri Mar 12 06:36:11 2010   it5              bills
      /acct/variables.fm
      Complete
  Fri Mar 12 06:36:11 2010   it5              bills
      /acct/securityMgmtSvcs.fm
      Complete
...
  Fri Mar 12 06:36:31 2010   budget           bills
      /acct/img/equipment/ACMschematic.pdf
      Complete
    Scan Statistics:
      Scan Started:                                  Fri Mar 12 06:36:08 2010
      Scan Completed:                                Fri Mar 12 06:36:31 2010
      Elapsed Time:                                  00:00:23
      Number of Times Paused:                        0
      Total Time Paused:                             00:00:00
      Number of Times Stopped by Low Space:          0
      Time Waiting for Free Space:                   00:00:00
      Files Scanned:                               1098
      Directories Scanned:                         155
      Files in Fileset:                            68
      Files Migrated:                              68
      Size of Files Migrated:                      89 MB
      Directories Promoted:                        0
      Failed Migrations:                           0
      Size of Failed Migrations:                   0 B
      Failed Directory Promotes:                   0
      Files Forced Closed:                         0
      Suppressed Error Messages:                   0

Total processed:              68
Elapsed time:           00:00:33
**** File Placement Report: DONE at Fri Mar 12 06:36:31 2010 ****
bstnA(gbl)# ...
```

Disabling Reports

From gbl-ns-vol-plc mode, use no report to stop generating placement reports for the current rule:

```
no report
```

For example, the following command sequence disables reporting for the rule, “mvTars:”

```
bstnA(gbl)# namespace archives volume /home
bstnA(gbl-ns-vol[archives~/home])# place-rule mvTars
bstnA(gbl-ns-vol-plc[archives~/home~mvTars])# no report
bstnA(gbl-ns-vol-plc[archives~/home~mvTars])# ...
```

Configuring Inline-Migration Reports

As mentioned above, some client events can cause a file to join a fileset between volume scans. These are called *inline* events. The place rule migrates the fileset to your chosen target as soon as the inline event occurs. From gbl-ns-vol-plc mode, you can use the inline report command to generate regular reports of all such inline migrations:

```
inline report {hourly | daily} prefix [verbose] [delete-empty |
error-only]
```

where

hourly | daily sets the frequency of these reports.

prefix (1-1024 characters) is the prefix to be used for this rule’s inline reports. Each report has a unique name in the following format: *prefixYearMonthDayHourMinute*.rpt (for example, inln_t2_200903031200.rpt for a report with the “inln_t2” prefix).

verbose (optional) includes one detailed line for each file.

delete-empty (optional) saves internal disk space by removing any inline-migration report without any errors or migrated files. Only reports with errors and/or migrated files remain.

error-only (optional) is another option for saving space. This deletes all reports without any errors. The only reports that remain are the ones that contain errors, whether or not they contain any migrated files.

For example, the following command sequence enables verbose inline-migration reports for the “busy2tier1” rule:

```
bstnA(gbl)# namespace medarcv volume /lab_equipment
bstnA(gbl-ns-vol[medarcv~/lab_equipment])# place-rule busy2tier1
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# inline report hourly leTier1 verbose
bstnA(gbl-ns-vol-plc[medarcv~/lab_equipment~busy2tier1])# ...
```

Finding and Showing Inline-Migration Reports

You can use the show reports type iPl command to list all inline-placement reports on the ARX. You can invoke this command from any mode:

show reports type iPl

Each report name contains a time stamp to help you identify the report you want. To show the report’s contents, use show reports *report-name* from any mode:

show reports *report-name*

For example, the following command sequence lists several inline-migration reports and then shows one of them from the “docsPlc” rule:

```
bstnA(gbl)# show reports type iPl


  reports
    Codes: iPl=Inline Place Rule
      docsPlc_201003120635.rpt Mar 12 07:03  2.8 kB        iPl  DONE: 6 in 00:59:59
      leTier1_201003120717.rpt Mar 12 08:17  2.1 kB        iPl  DONE: 0 in 00:59:59

bstnA(gbl)# show reports docsPlc_20120411004754.rpt
**** File Placement Report: Started at 04/11/2012 00:47:54 -0400 ****
**** Software Version: 6.02.000.14353 (Apr  6 2012 20:12:43) [nbuilds]
**** Hardware Platform: ARX-4000
**** Report Destination:

  Place Rule:       docs2das8

    Configuration:
      Namespace:                                     wwmed
      Volume Name:                                   /acct
      From fileset:                                  bulky (files only)
      Target share:                                  bills
      Report:                                        docsPlc
      Report Verbose:                                Enabled
      Report Delete Empty:                           Disabled
      Report Errors Only:                            Disabled
      Inline Report:                                 docsPlc
      Inline Report Interval:                        hourly
      Inline Report Verbose:                         Enabled
      Inline Report Delete Empty:                    Disabled
      Inline Report Errors Only:                     Disabled
      Migrate limit:                                 50G
      Volume Scan:                                   Enabled
      Inline Notifications:                          Enabled
      Promote Directories:                           Disabled
      Auto-Close Files:                              Disabled
      Migrate Hard Links:                            Disabled

      Tentative:                                     No
      State:                                         Enabled


  Date                       Source Share     Target Share
      File
      Result
  --------------------------------------------------------------------------
  04/11/2012 00:48:06 -0400  it5              bills
      /acct/shellCmds_winXP.fm
      Complete
  04/11/2012 00:48:07 -0400  it5              bills
      /acct/advancedUser.fm
      Complete
  04/11/2012 00:48:07 -0400  it5              bills
      /acct/intro.fm
      Complete
  04/11/2012 00:48:07 -0400  it5              bills
      /acct/security_kerberos_cfg.fm
      Complete
  04/11/2012 00:48:07 -0400  it5              bills
      /acct/usersGroups.fm
      Complete
  04/11/2012 00:48:07 -0400  it5              bills
      /acct/layer3.fm
      Complete
  04/11/2012 00:48:07 -0400  it5              bills
      /acct/swUpgrade.fm
      Complete
  04/11/2012 00:48:07 -0400  it5              bills
      /acct/connectathon.fm
      Complete
  04/11/2012 00:48:07 -0400  it5              bills
      /acct/ppp_cfg.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/findShare.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/productOverview.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/services.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/backupRestore.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/troubleShoot.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/filesetPolicy.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/rework_vpn.fm
      Complete
  04/11/2012 00:48:08 -0400  bills2           bills
```

```
      /acct/payable/Glossary.pdf
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/troubleShootNet.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/nasStorageCheckList.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/netL2.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/nsCheck.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/HighAvail.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/costsIX.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/fileset.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/shellCmds_bash.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/globalWAN.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/preso_net_cfg.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/layer2.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/troubleShootNas8.x.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/securityNfs.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/variables.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/securityMgmtSvcs.fm
      Complete
  04/11/2012 00:48:08 -0400  bills2           bills
      /acct/payable/FrontMatter_bills2-3.fm
      Complete
  04/11/2012 00:48:08 -0400  bills2           bills
      /acct/payable/variables_bills2-3.fm
      Complete
  04/11/2012 00:48:08 -0400  bills2           bills
      /acct/payable/howTo.fm
      Complete
  04/11/2012 00:48:08 -0400  it5              bills
      /acct/FrontMatter.fm
      Complete
  04/11/2012 00:48:08 -0400  bills2           bills
      /acct/payable/TOC.fm
      Complete
```

```
04/11/2012 00:48:08 -0400  it5              bills
    /acct/cliOperator.pdf
    Complete
04/11/2012 00:48:12 -0400  budget           bills
    /acct/images/BookCover.eps
    Complete
04/11/2012 00:48:13 -0400  bills2           bills
    /acct/planner/downloads/gdb-refcard-letter.pdf
    Complete
04/11/2012 00:48:14 -0400  budget           bills
    /acct/production/relNotes/alpha1.pdf
    Complete
04/11/2012 00:48:17 -0400  budget           bills
    /acct/stevens/code/fileIO/cliOperator.pdf
    Complete
04/11/2012 00:48:20 -0400  it5              bills
    /acct/docs/masterIndex/masterBookIX.pdf
    Complete
04/11/2012 00:48:21 -0400  it5              bills
    /acct/docs/Quickstart_Card/HW_quickstart.pdf
    Complete
04/11/2012 00:48:21 -0400  it5              bills
    /acct/docs/sitePlanning/sitePlanning.pdf
    Complete
04/11/2012 00:48:21 -0400  it5              bills
    /acct/docs/snmpRef/snmpRef.pdf
    Complete
04/11/2012 00:48:21 -0400  it5              bills
    /acct/docs/logCatalog/logCatalog.pdf
    Complete
04/11/2012 00:48:21 -0400  it5              bills
    /acct/docs/cliReference/cliReference.pdf
    Complete
04/11/2012 00:48:21 -0400  it5              bills
    /acct/docs/cliOperator/cliOperator.pdf
    Complete
04/11/2012 00:48:21 -0400  it5              bills
    /acct/docs/SLM_Beta/SLM.book.pdf
    Complete
04/11/2012 00:48:23 -0400  it5              bills
    /acct/docs/GUI/GUI.pdf
    Complete
04/11/2012 00:48:23 -0400  it5              bills
    /acct/docs/ReleaseNotes/notes.pdf
    Complete
04/11/2012 00:48:23 -0400  it5              bills
    /acct/docs/Quickstart_CardA1K/A1K_quickstart.pdf
    Complete
04/11/2012 00:48:23 -0400  it5              bills
    /acct/docs/Quickstart_CardA1K/A1Kplus_quickstart.pdf
    Complete
04/11/2012 00:48:23 -0400  it5              bills
    /acct/docs/HW_InstallA5C/HWInstall.pdf
    Complete
04/11/2012 00:48:24 -0400  it5              bills
    /acct/docs/useCases/multiTier/multiTier.pdf
    Complete
04/11/2012 00:48:24 -0400  it5              bills
    /acct/docs/useCases/network/network.pdf
    Complete
04/11/2012 00:48:26 -0400  it5              bills
    /acct/docs/useCases/singleTier/singleTier.pdf
```

```
      Complete
  04/11/2012 00:48:26 -0400  it5              bills
      /acct/docs/useCases/specialCases/specialCases.pdf
      Complete
  04/11/2012 00:48:26 -0400  it5              bills
      /acct/docs/HW_Install/HW_Book/HWInstall.pdf
      Complete
  04/11/2012 00:48:26 -0400  it5              bills
      /acct/docs/HW_InstallA1K/HW_Book/HWInstall.pdf
      Complete
  04/11/2012 00:48:26 -0400  it5              bills
      /acct/docs/SecureAgent/Framefiles/SecureAgent.pdf
      Complete
  04/11/2012 00:48:26 -0400  it5              bills
      /acct/docs/Glossary/Frame_files/Glossary.pdf
      Complete
  04/11/2012 00:48:26 -0400  budget           bills
      /acct/images/a6000/ASM_FC.tif
      Complete
  04/11/2012 00:48:26 -0400  budget           bills
      /acct/images/a6000/NSM_fiber.tif
      Complete
  04/11/2012 00:48:26 -0400  budget           bills
      /acct/img/equipment/APMschematic.pdf
      Complete
  04/11/2012 00:48:26 -0400  budget           bills
      /acct/img/equipment/ACMschematic.pdf
      Complete
  04/11/2012 00:48:26 -0400  budget           bills
      /acct/img/equipment/SCMschematic.pdf
      Complete
    Scan Statistics:
      Scan Started:                                  04/11/2012 00:48:05 -0400
      Scan Completed:                                04/11/2012 00:48:26 -0400
      Elapsed Time:                                  00:00:21
      Number of Times Paused:                        0
      Total Time Paused:                             00:00:00
      Number of Times Stopped by Low Space:          0
      Time Waiting for Free Space:                   00:00:00
      Files Scanned:                                 1211
      Directories Scanned:                           165
      Files in Fileset:                              73
      Files Migrated:                                68
      Size of Files Migrated:                        89 MB
      Directories Promoted:                          0
      Failed Migrations:                             0
      Size of Failed Migrations:                     0 B
      Failed Directory Promotes:                     0
      Files Forced Closed:                           0
      Suppressed Error Messages:                     0

Total processed:              68
Elapsed time:           00:00:32
**** File Placement Report: DONE at 04/11/2012 00:48:26 -0400 ****

bstnA(gbl)# ...
```

Disabling Inline Migration Reports

From gbl-ns-vol-plc mode, use no inline report to stop generating inline-migration reports for the current rule:

```
no inline report
```

For example, the following command sequence disables inline-migration reporting for the rule, “masterDirs2Rx:”

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# place-rule masterDirs2Rx
bstnA(gbl-ns-vol-plc[medarcv~/rcrds~masterDirs2Rx])# no inline report
bstnA(gbl-ns-vol-plc[medarcv~/rcrds~masterDirs2Rx])# ...
```

Making the Rule Tentative

A *tentative* rule is configured to appear in the system logs (syslog) as “tentative,” showing the potential effects of the rule if it was enabled. (The log component, POLICY_ACTION, creates the syslog messages; syslog access and log components are described in the *ARX CLI Maintenance Guide*.) If you configured reporting for the rule (as shown above), the report shows which files would be migrated if the rule was fully enabled. Tentative rules do not change policy enforcement, but they do consume processing time in the policy software. From gbl-ns-vol-plc mode, use the tentative command to put the placement rule in a tentative state:

```
tentative
```

To see the simulated rule run, you must enable the rule (as described below). Then use show logs syslog or grep *pattern* logs syslog to view the syslog.

For example, the following command sequence makes the “mvTars” rule tentative:

```
bstnA(gbl)# namespace archives volume /home
bstnA(gbl-ns-vol[archives~/home])# place-rule mvTars
bstnA(gbl-ns-vol-plc[archives~/home~mvTars])# tentative
bstnA(gbl-ns-vol-plc[archives~/home~mvTars])# ...
```

Removing the Tentative State

Use the no tentative command to fully activate the file-placement rule:

```
no tentative
```

This activates the rule so that actual migrations occur when the rule is next enabled (as described below) and/or the next time the rule’s schedule fires.

For example, the following command sequence activates the “docs2das8” rule:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# no tentative
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

Enabling the Placement Rule

The final step in configuring any rule is to enable it. By default, the rule is disabled and ignored by policy software. Use the enable command to enable the rule.

enable

For example, the following command sequence enables the "docs2das8" rule:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# enable
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

Disabling the Rule

Disabling the rule removes it from consideration. Use no enable from gbl-ns-vol-plc mode to disable a placement rule.

no enable

For example, the following command sequence disables the "docs2das8" rule:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# place-rule docs2das8
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# no enable
bstnA(gbl-ns-vol-plc[wwmed~/acct~docs2das8])# ...
```

Showing the Effects of File Placement

A file-placement rule moves files and/or directories to new shares. To show the effects of the file-placement rule, you can run an nsck report: nsck *namespace* metadata-only shows file/directory locations on back-end shares

You can invoke this command from priv-exec mode. It generates a report file that you can see with show reports. The metadata-only report is named "metadata_only.*id*.rpt" by default, where *id* identifies the nsck job. Use show, tail, or grep to view the contents of the report. See *Showing Metadata*, on page 7-8 of the *ARX CLI Maintenance Guide* for full details on creating a metadata-only report.

Removing the Placement Rule

You can remove a placement rule to both disable it and delete its configuration. Many file-placement rules can manipulate directory mastership so that directory trees grow naturally on desired filers. If all directory masters are placed correctly, the managed volume creates new files and directories under them by default; the file-placement rule is no-longer needed.

Use the no form of the place-rule command to remove a placement rule:

no place-rule *name*

where ***name*** (1-64 characters) identifies the rule to be removed.

For example, the following command sequence removes the placement rule, “dirsToNAS19,” from the “medarcv~/rcrds” volume:

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# no place-rule dirsToNAS19
bstnA(gbl-ns-vol[medarcv~/rcrds])# ...
```

If the rule matches files and/or directories based on their names and/or ages instead of their paths, you may want to keep the rule indefinitely. The file-placement rule can continue to match against newly-created files/directories, steering them as needed. It can also re-assess existing files/directories as they age, or as clients change them, and migrate them as needed.

Showing All Rules

Use the show policy command to get a full list of all file-placement rules on the system, along with any other policy objects:

show policy

This shows a list of rules and share farms grouped by namespace and volume.

For example:

```
bstnA# show policy

                                                                            Status
Namespace:             medco
Volume:                /vol

Rule                      Type              Status
------------------------  ----------------  ----------------------------------------------

Namespace:             wwmed
Volume:                /acct

Rule                      Type              Status
------------------------  ----------------  ----------------------------------------------
docs2das8                 Place             Vol. Scan: Initializing   Migration: Initializing
fm1                       Share Farm        Vol. Scan: Complete      Migration: Complete

Namespace:             medarcv
Volume:                /rcrds

Rule                      Type              Status
------------------------  ----------------  ----------------------------------------------
rcrdsArchive              Snapshot          Enabled
dailyArchive              Place             Vol. Scan: Complete      Migration: Complete
masterDirs2Rx             Place             Vol. Scan: Complete      Migration: Complete
medFm                     Share Farm        Vol. Scan: Complete      Migration: Complete

Namespace:             medarcv
Volume:                /lab_equipment

Rule                      Type              Status
------------------------  ----------------  ----------------------------------------------
hourlySnap                Snapshot          Enabled
dailySnap                 Snapshot          Enabled
mirrorSnap                Replica Snapshot  Enabled
labArchive                Snapshot          Enabled
busy2tier1                Place             Vol. Scan: Paused        Migration: Paused
nonbusy2tier2             Place             Vol. Scan: Paused        Migration: Paused
masterDirs2Tier1          Place             Vol. Scan: Complete      Migration: Complete
tier1                     Share Farm        Vol. Scan: Complete      Migration: Complete
tier2                     Share Farm        Vol. Scan: Complete      Migration: Complete

Namespace:             medarcv
Volume:                /test_results

Rule                      Type              Status
------------------------  ----------------  ----------------------------------------------

Namespace:             medarcv
Volume:                /acopia$ns3
```

```
Rule                      Type              Status
------------------------  ----------------  ----------------------------------------------

Namespace:             insur
Volume:                /claims

Rule                      Type              Status
------------------------  ----------------  ----------------------------------------------
mpHourlySnap              Snapshot          Enabled

Namespace:             insur
Volume:                /acopia$ns4


bstnA# ...
```

Showing Details

Add the details keyword to the end of the command to show details for all rules, schedules, and share farms on the ARX:

- details

The output is divided into namespaces, volumes, and rules. All namespaces and volumes are listed, even those without any rules or share farms. Each rule has one or more tables describing its configuration, its latest status, and various statistics. Share farms appear at the same level as rules. For example:

```
bstnA# show policy details

Namespace:             medco

  Namespace Migration Configuration:
    Migrate-Attempts:                              32
    Migrate-Delay:                                 30 Seconds
    Migrate-Retry-Delay:                           900 Seconds

Volume:                /vol

  Volume Snapshot Configuration:
    Point-in-Time Consistency:                     Disabled
    Management Command Timeout:                    80 seconds
    NFS Directory Name:                            .snapshot
    Directory Display:                             None

Namespace:             wwmed

  Namespace Migration Configuration:
    Migrate-Attempts:                              32
    Migrate-Delay:                                 30 Seconds
    Migrate-Retry-Delay:                           900 Seconds

Volume:                /acct

  Volume Snapshot Configuration:
    Point-in-Time Consistency:                     Disabled
    Management Command Timeout:                    80 seconds
    NFS Directory Name:                            .snapshot
    Directory Display:                             None

  Share Farm:       fm1

                  Placement Frequency   Freespace Status
     Share Name   Count/Total           Free   Size   Pct Free
     -----------  --------------------  ----------------------
     bills        96  / 766      12 %   27 GB   70 GB   39 %
     budget       282 / 766      36 %   81 GB   89 GB   91 %
     bills2       388 / 766      50 %   111 GB  113 GB  98 %


  New File Placement Rule:  fm1

    Configuration:
      Constrain Files:                               No
      Constrain Directories:                         No
      Balance Mode:                                  Capacity
      Auto Migrate:                                  Enabled

      State:                                         Enabled

    Status:
      Volume Scan Status:                            Complete
      File Migration Status:                         Complete
      New File Placement Status:                     Enabled

    Inline Statistics:
      Inline Interval Start:                         03/26/2012 00:51:51 -0400
      Inline Interval Elapsed:                       00:53:03
      Inline Files Migrated:                         0
      Inline Failed Migrations:                      0
      Inline Size of Failed Migrations:              0 B
      Inline Files Renamed:                          0
      Inline Files Placed:                           0
      Inline Directories Renamed:                    0
      Inline Directories Placed:                     0
      Inline Directories Promoted:                   0
      Inline Failed Directory Promotes:              0

    Cumulative Statistics:
      Total Files Migrated From Scan:                0
      Total Files Migrated From Inline Activity:     0
      Total Directories Promoted:                    0
      Total Failed Migrations:                       0
      Total Size of Failed Migrations:               0 B
      Total Failed Directory Promotes:               0
      Total Files Forced Closed:                     0
      Total Retried Migrations:                      0
      Total Canceled Migrations:                     0
      Total Hard Links Skipped:                      0
      Total Files Placed Inline:                     0
      Total File Renames Processed Inline:           0
      Total Directories Placed Inline:               0
      Total Directory Renames Processed Inline:      0
      Number of Inline Overflow Errors:              0
      Number of Scans Performed:                     0

    Queue Statistics:
      First-time Migrates:                           0
      Requeued Migrates:                             0
      Queued Directory Promotes:                     0
```

```
Place Rule:       docs2das8

  Configuration:
    From fileset:                                  bulky (files only)
    Target share:                                  bills
    Report:                                        docsPlc
    Report Verbose:                                Enabled
    Report Delete Empty:                           Disabled
    Report Errors Only:                            Disabled
    Inline Report:                                 docsPlc
    Inline Report Interval:                        hourly
    Inline Report Verbose:                         Enabled
    Inline Report Delete Empty:                    Disabled
    Inline Report Errors Only:                     Disabled
    Migrate limit:                                 50G
    Volume Scan:                                   Enabled
    Inline Notifications:                          Enabled
    Promote Directories:                           Disabled
    Auto-Close Files:                              Disabled
    Migrate Hard Links:                            Disabled

    Tentative:                                     No
    State:                                         Enabled

  Status:
    Volume Scan Status:                            Initializing
    File Migration Status:                         Initializing
    New File Placement Status:                     Initializing

  Inline Statistics:
    Inline Interval Start:                         03/26/2012 00:51:49 -0400
    Inline Interval Elapsed:                       00:53:05
    Inline Files Migrated:                         0
    Inline Failed Migrations:                      0
    Inline Size of Failed Migrations:              0 B
    Inline Files Renamed:                          3
    Inline Files Placed:                           33
    Inline Directories Renamed:                    0
    Inline Directories Placed:                     0
    Inline Directories Promoted:                   0
    Inline Failed Directory Promotes:              0

  Cumulative Statistics:
    Total Files Migrated From Scan:                0
    Total Files Migrated From Inline Activity:     0
    Total Directories Promoted:                    0
    Total Failed Migrations:                       0
    Total Size of Failed Migrations:               0 B
    Total Failed Directory Promotes:               0
    Total Files Forced Closed:                     0
    Total Retried Migrations:                      0
    Total Canceled Migrations:                     0
    Total Hard Links Skipped:                      0
    Total Files Placed Inline:                     33
    Total File Renames Processed Inline:           3
    Total Directories Placed Inline:               0
    Total Directory Renames Processed Inline:      0
    Number of Inline Overflow Errors:              0
```

```
      Number of Scans Performed:                    0

    Queue Statistics:
      First-time Migrates:                          0
      Requeued Migrates:                            0
      Queued Directory Promotes:                    0



Namespace:             medarcv

  Namespace Migration Configuration:
    Migrate-Attempts:                             32
    Migrate-Delay:                                30 Seconds
    Migrate-Retry-Delay:                          900 Seconds

Volume:                /rcrds

  Volume Snapshot Configuration:
    Point-in-Time Consistency:                    Disabled
    Management Command Timeout:                   80 seconds
    CIFS Directory Name:                          ~snapshot
    Directory Display:                            None
    Hidden File Attribute:                        No
    Restricted Access Configured:                 No
    VSS Mode:                                     Windows XP

  Share Farm:       medFm

                  Placement Frequency   Freespace Status
     Share Name   Count/Total           Free   Size   Pct Free
     -----------  --------------------  ----------------------
     rx           63  / 147      42 %   875 MB  1.9 GB  42 %
     charts       84  / 147      57 %   1.3 GB  1.9 GB  69 %


...
bstnA# ...
```

Focusing on One Namespace

You can add the namespace clause to focus on a particular namespace:

show policy namespace *namespace*

where ***namespace*** (optional, 1-30 characters) is the name of a configured namespace (see *Listing All Namespaces*, on page 7-4).

This lists all rules in the chosen namespace, and shows the rule order. Rule order is important for any rules that may conflict: if rule 1 contradicts rule 4 for a given file, rule 1 is enforced for that file.

For example, the following command lists the rule and share farm for the “wwmed” namespace:

```
bstnA# show policy namespace wwmed

Namespace:             wwmed

  Namespace Migration Configuration:
    Migrate-Attempts:                            32
    Migrate-Delay:                               30 Seconds
    Migrate-Retry-Delay:                         900 Seconds

Volume:             /acct

  Rule
Priority   Rule                      Type             Status
---------  ------------------------  ----------------  ---------------------------------------
    1      docs2das8                 Place            Vol. Scan: Complete   Migration: Complete
    2      fm1                       Share Farm       Vol. Scan: Complete   Migration: Complete

bstnA# ...
```

Showing Details for the Namespace

Add the details keyword for details about the namespace:

show policy namespace *namespace* details

This lists all the volumes in the namespace, with details about the rules and share farms under each volume. Volumes without any policy settings are listed without any details under them.

For example, this command shows details about the "wwmed" namespace:

```
bstnA# show policy namespace wwmed details

Namespace:          wwmed

  Namespace Migration Configuration:
    Migrate-Attempts:                            32
    Migrate-Delay:                               30 Seconds
    Migrate-Retry-Delay:                         900 Seconds

Volume:             /acct

  Volume Snapshot Configuration:
    Point-in-Time Consistency:                   Disabled
    Management Command Timeout:                  80 seconds
    NFS Directory Name:                          .snapshot
    Directory Display:                           None

  Share Farm:      fm1

                 Placement Frequency   Freespace Status
     Share Name  Count/Total           Free   Size   Pct Free
     ----------- --------------------  ---------------------
     bills       96  / 766      12 %   27 GB   70 GB   39 %
     budget      282 / 766      36 %   81 GB   89 GB   91 %
     bills2      388 / 766      50 %   111 GB  113 GB  98 %


  New File Placement Rule:  fm1

    Configuration:
      Constrain Files:                             No
      Constrain Directories:                       No
      Balance Mode:                                Capacity
      Auto Migrate:                                Enabled

      State:                                       Enabled
    Status:
      Volume Scan Status:                            Complete
      File Migration Status:                         Complete
      New File Placement Status:                     Enabled

    Inline Statistics:
      Inline Interval Start:                         03/26/2012 00:51:51 -0400
      Inline Interval Elapsed:                       00:53:05
      Inline Files Migrated:                         0
      Inline Failed Migrations:                      0
      Inline Size of Failed Migrations:              0 B
      Inline Files Renamed:                          0
      Inline Files Placed:                           0
      Inline Directories Renamed:                    0
      Inline Directories Placed:                     0
      Inline Directories Promoted:                   0
      Inline Failed Directory Promotes:              0

    Cumulative Statistics:
      Total Files Migrated From Scan:                0
      Total Files Migrated From Inline Activity:     0
      Total Directories Promoted:                    0
      Total Failed Migrations:                       0
      Total Size of Failed Migrations:               0 B
      Total Failed Directory Promotes:               0
      Total Files Forced Closed:                     0
      Total Retried Migrations:                      0
      Total Canceled Migrations:                     0
      Total Hard Links Skipped:                      0
      Total Files Placed Inline:                     0
      Total File Renames Processed Inline:           0
      Total Directories Placed Inline:               0
      Total Directory Renames Processed Inline:      0
      Number of Inline Overflow Errors:              0
      Number of Scans Performed:                     0

    Queue Statistics:
      First-time Migrates:                           0
      Requeued Migrates:                             0
      Queued Directory Promotes:                     0



  Place Rule:       docs2das8

    Configuration:
      From fileset:                                  bulky (files only)
      Target share:                                  bills
      Report:                                        docsPlc
      Report Verbose:                                Enabled
      Report Delete Empty:                           Disabled
      Report Errors Only:                            Disabled
      Inline Report:                                 docsPlc
      Inline Report Interval:                        hourly
      Inline Report Verbose:                         Enabled
      Inline Report Delete Empty:                    Disabled
      Inline Report Errors Only:                     Disabled
      Migrate limit:                                 50G
      Volume Scan:                                   Enabled
      Inline Notifications:                          Enabled
```

```
      Promote Directories:                           Disabled
      Auto-Close Files:                              Disabled
      Migrate Hard Links:                            Disabled

      Tentative:                                     No
      State:                                         Enabled

    Status:
      Volume Scan Status:                            Initializing
      File Migration Status:                         Initializing
      New File Placement Status:                     Initializing

    Inline Statistics:
      Inline Interval Start:                         03/26/2012 00:51:49 -0400
      Inline Interval Elapsed:                       00:53:07
      Inline Files Migrated:                         0
      Inline Failed Migrations:                      0
      Inline Size of Failed Migrations:              0 B
      Inline Files Renamed:                          3
      Inline Files Placed:                           33
      Inline Directories Renamed:                    0
      Inline Directories Placed:                     0
      Inline Directories Promoted:                   0
      Inline Failed Directory Promotes:              0

    Cumulative Statistics:
      Total Files Migrated From Scan:                0
      Total Files Migrated From Inline Activity:     0
      Total Directories Promoted:                    0
      Total Failed Migrations:                       0
      Total Size of Failed Migrations:               0 B
      Total Failed Directory Promotes:               0
      Total Files Forced Closed:                     0
      Total Retried Migrations:                      0
      Total Canceled Migrations:                     0
      Total Hard Links Skipped:                      0
      Total Files Placed Inline:                     33
      Total File Renames Processed Inline:           3
      Total Directories Placed Inline:               0
      Total Directory Renames Processed Inline:      0
      Number of Inline Overflow Errors:              0
      Number of Scans Performed:                     0

    Queue Statistics:
      First-time Migrates:                           0
      Requeued Migrates:                             0
      Queued Directory Promotes:                     0
...

bstnA# ...
```

Focusing on One Volume

Add the volume clause to focus on one of the namespace’s volumes:

show policy namespace *namespace* volume *volume*

where

namespace (optional, 1-30 characters) is the namespace, and

volume (optional, 1-1024 characters) identifies the volume.

The output shows some namespace-configuration settings, volume snapshot settings, free space settings and status for the volume's shares, and a table of the volume's rules.

For example, the following command shows the "/acct" volume in the "wwmed" namespace:

```
bstnA(gbl)# show policy namespace wwmed volume /acct

Namespace:            wwmed

  Namespace Migration Configuration:
    Migrate-Attempts:                            32
    Migrate-Delay:                               30 Seconds
    Migrate-Retry-Delay:                         900 Seconds

Volume:               /acct

  Volume Snapshot Configuration:
    Point-in-Time Consistency:                   Disabled
    Management Command Timeout:                  80 seconds
    NFS Directory Name:                          .snapshot
    Directory Display:                           None

  Share Freespace:

                 Free Space Thresholds   Freespace Status
    Share Name   Maintain    Resume      Free   Size    Pct Free
    -----------  ---------------------  ----------------------
    budget            2 %        3 %      81 GB   89 GB   91%
    bills             2 %        3 %      27 GB   70 GB   39%
    bills2            2 %        3 %     111 GB  113 GB   98%
    it5               2 %        3 %      46 GB   56 GB   82%

  Rule
Priority   Rule                      Type             Status
---------  ------------------------  ----------------  ---------------------------------------
    1      docs2das8                 Place            Vol. Scan: Initializing   Migration:
Initializing
    2      fm1                       Share Farm       Vol. Scan: Complete   Migration: Complete

bstnA# ...
```

Showing Details for the Volume

As with namespaces, you can add the details keyword for details about the volume:

show policy namespace *namespace* volume *volume* details

This lists details about all the rules and share farms in the volume. For example, this command shows details about the "wwmed~/acct" volume:

```
bstnA# show policy namespace wwmed volume /acct details

Namespace:            wwmed

  Namespace Migration Configuration:
    Migrate-Attempts:                            32
    Migrate-Delay:                               30 Seconds
    Migrate-Retry-Delay:                         900 Seconds
```

```
Volume:                /acct

  Volume Snapshot Configuration:
    Point-in-Time Consistency:                     Disabled
    Management Command Timeout:                    80 seconds
    NFS Directory Name:                            .snapshot
    Directory Display:                             None

  Share Farm:       fm1

                  Placement Frequency   Freespace Status
     Share Name   Count/Total           Free   Size   Pct Free
     -----------  --------------------  ---------------------
     bills        95  / 766      12 %   27 GB   70 GB   39 %
     budget       283 / 766      36 %   81 GB   89 GB   91 %
     bills2       388 / 766      50 %   111 GB  113 GB  98 %


  New File Placement Rule:  fm1

    Configuration:


...
```

Focusing on One Share Farm or Rule

After the optional namespace and volume clauses, you can use the rule keyword with the name of a rule or share farm. This narrows the focus to one object in the volume:

show policy namespace ***namespace*** **volume** ***volume*** **rule** ***rule-or-farm-name***

where

namespace (optional, 1-30 characters) is the namespace,

volume (optional, 1-1024 characters) identifies the volume, and

rule-or-farm-name (optional, 1-1024 characters) identifies the rule or share farm.

This expands the output to show the full details and statistics for the share farm or rule. These details include configuration parameters and usage statistics.

For example, the following command shows the “docs2das8” rule in the “wwmed~/acct” volume:

```
bstnA# show policy namespace wwmed volume /acct rule docs2das8

Namespace:             wwmed

  Namespace Migration Configuration:
    Migrate-Attempts:                              32
    Migrate-Delay:                                 30 Seconds
    Migrate-Retry-Delay:                           900 Seconds

Volume:                /acct
```

```
Volume Snapshot Configuration:
  Point-in-Time Consistency:                     Disabled
  Management Command Timeout:                    80 seconds
  NFS Directory Name:                            .snapshot
  Directory Display:                             None

Place Rule:       docs2das8

  Configuration:
    From fileset:                                  bulky (files only)
    Target share:                                  bills
    Report:                                        docsPlc
    Report Verbose:                                Enabled
    Report Delete Empty:                           Disabled
    Report Errors Only:                            Disabled
    Inline Report:                                 docsPlc
    Inline Report Interval:                        hourly
    Inline Report Verbose:                         Enabled
    Inline Report Delete Empty:                    Disabled
    Inline Report Errors Only:                     Disabled
    Migrate limit:                                 50G
    Volume Scan:                                   Enabled
    Inline Notifications:                          Enabled
    Promote Directories:                           Disabled
    Auto-Close Files:                              Disabled
    Migrate Hard Links:                            Disabled

    Tentative:                                     No
    State:                                         Enabled

  Status:
    Volume Scan Status:                            Initializing
    File Migration Status:                         Initializing
    New File Placement Status:                     Initializing

  Inline Statistics:
    Inline Interval Start:                         03/26/2012 00:51:49 -0400
    Inline Interval Elapsed:                       00:53:08
    Inline Files Migrated:                         0
    Inline Failed Migrations:                      0
    Inline Size of Failed Migrations:              0 B
    Inline Files Renamed:                          3
    Inline Files Placed:                           33
    Inline Directories Renamed:                    0
    Inline Directories Placed:                     0
    Inline Directories Promoted:                   0
    Inline Failed Directory Promotes:              0

  Cumulative Statistics:
    Total Files Migrated From Scan:                0
    Total Files Migrated From Inline Activity:     0
    Total Directories Promoted:                    0
    Total Failed Migrations:                       0
    Total Size of Failed Migrations:               0 B
    Total Failed Directory Promotes:               0
    Total Files Forced Closed:                     0
    Total Retried Migrations:                      0
    Total Canceled Migrations:                     0
    Total Hard Links Skipped:                      0
    Total Files Placed Inline:                     33
    Total File Renames Processed Inline:           3
    Total Directories Placed Inline:               0
```

```
      Total Directory Renames Processed Inline:      0
      Number of Inline Overflow Errors:              0
      Number of Scans Performed:                     0

    Queue Statistics:
      First-time Migrates:                           0
      Requeued Migrates:                             0
      Queued Directory Promotes:                     0

bstnA# ...
```

Sample Configuration - Age-Based Tiering

Consider a site with two CIFS servers, where one server is faster and more reliable than the other. This example presumes that a managed volume, “swic~/users,” contains a share from each of these servers. The example uses file placement rules to establish tiered storage: the share on the superior server, “win1,” stores all the files that have been modified in the past 30 days, and the other share (“samba1”) holds the remainder of the files.

This sample begins by creating a daily schedule:

```
bstnA(gbl)# schedule every_day
bstnA(gbl-schedule[every_day])# every 1 days
bstnA(gbl-schedule[every_day])# start 11/12/2009:01:00:00
bstnA(gbl-schedule[every_day])# exit
bstnA(gbl)#
```

The next command sequence creates two age-based filesets, one for files newer than 30 days and another for files older than 30 days.

```
bstnA(gbl)# policy-age-fileset active_files

This will create a new policy object.

Create object 'active_files'? [yes/no]   yes
bstnA(gbl-fs-age[active_files])# select-files newer-than 30 days
bstnA(gbl-fs-age[active_files])# exit
bstnA(gbl)# policy-age-fileset dormant_files

This will create a new policy object.

Create object 'dormant_files'? [yes/no]   yes
bstnA(gbl-fs-age[dormant_files])# select-files older-than 30 days
bstnA(gbl-fs-age[dormant_files])# exit
bstnA(gbl)#
```

The final command sequence creates two file-placement rules. The first rule sends the new files to Tier 1 (the “win1” share), and the second rule sends the older files to the Tier-2 share. Note that inline notifications are disabled in the Tier-2 rule only.

```
bstnA(gbl)# namespace swic volume /users
bstnA(gbl-ns-vol[swic~/users])# place-rule active_to_new_srvrs

This will create a new policy object.

Create object 'active_to_new_srvrs'? [yes/no]   yes
bstnA(gbl-ns-vol-plc[swic~/users~active_to_new_srvrs])# from fileset active_files
bstnA(gbl-ns-vol-plc[swic~/users~active_to_new_srvrs])# target share win1
bstnA(gbl-ns-vol-plc[swic~/users~active_to_new_srvrs])# schedule every_day
bstnA(gbl-ns-vol-plc[swic~/users~active_to_new_srvrs])# enable
bstnA(gbl-ns-vol-plc[swic~/users~active_to_new_srvrs])# exit
bstnA(gbl-ns-vol-plc[swic~/users])# place-rule dormant_to_old_srvr

This will create a new policy object.

Create object 'dormant_to_old_srvr'? [yes/no]   yes
bstnA(gbl-ns-vol-plc[swic~/users~dormant_to_old_srvr])# from fileset dormant_files
bstnA(gbl-ns-vol-plc[swic~/users~dormant_to_old_srvr])# target share samba1
bstnA(gbl-ns-vol-plc[swic~/users~dormant_to_old_srvr])# no inline notify
bstnA(gbl-ns-vol-plc[swic~/users~dormant_to_old_srvr])# schedule every_day
bstnA(gbl-ns-vol-plc[swic~/users~dormant_to_old_srvr])# enable
bstnA(gbl-ns-vol-plc[swic~/users~dormant_to_old_srvr])# exit
bstnA(gbl-ns-vol[swic~/users])# exit
bstnA(gbl-ns[swic])# exit
bstnA(gbl)# ...
```

With these rules in place, the policy engine examines the volume at 1AM every morning and migrates files as needed. The policy engine performs a single volume scan for both placement rules, since both rules use the "every_day" schedule. The scan divides all of the volume's files into two filesets: one with files modified less than 30 days ago, and another with all the rest of the files in the volume. If any file is on the wrong tier at 1AM, the policy engine migrates it to the correct one.

Inline notifications are enabled for the Tier-1 rule, "active_to_new_srvrs," so the files migrate as clients change them. Therefore, most newly-modified rules are already on Tier 1 before 1AM. Most of the migrations at 1AM are from Tier 1 to Tier 2, for files that have aged out beyond 30 days.

The assessment of "30 days old" comes during the 1AM volume scan. A file that is modified on September 4 is 14 days old on September 18 and 29 days old on October 3. The file migrates to Tier 2 at 1AM on October 5, when it is 31 days old.

Changing Rule Order

The policy software enforces its rules in order. Whenever two rules conflict, the higher-order rule is enforced and the lower-order rule is not. Conflicts arise for files and/or directories that match two different rules; each rule may attempt to place the file or directory on a different target. By default, rules are ordered on a first-come-first-served basis; the first rule you enter is of the highest order, and the last rule is the lowest order.

You can change the rule order only for placement rules that use filesets as their sources. These are the types of rules described in this chapter. The priority cannot change for shadow-copy rules (which are always highest priority), placement rules that drain shares (which are second priority), and share farms (which are lowest priority). Fileset-placement rules are grouped together between the second and last priority groups; they are lower-priority than drain rules, but they take precedence over all share-farm rules.

Use the policy order-rule command in gbl-ns-vol mode to change the rule order for fileset-placement rules:

```
policy order-rule rule1 {before | after} rule2
```

where

rule1 (1-1024 characters) identifies a rule to move,

before | after is a required choice to set the new position for *rule1*, and

rule2 (1-64 characters) identifies a second rule, whose order stays the same.

If *rule1* has no schedule, it immediately migrates all files where *rule2* previously guided their placement.

Use the show policy namespace command to see the current rule order for a namespace. Refer back to *Focusing on One Namespace*, on page 14-38. If ***rule1*** already precedes ***rule2***, you cannot place it **before** rule2. Conversely, you cannot place ***rule1*** **after** ***rule2*** if it is already after it.

For example, the following command sequence shows the rule order in the “wwmed” namespace and then puts a file-placement rule, “placeTest,” directly after the “docs2das8” rule:

```
bstnA(gbl)# show policy namespace wwmed

Namespace:             wwmed

  Namespace Migration Configuration:
    Migrate-Attempts:                          32
    Migrate-Delay:                             30 Seconds
    Migrate-Retry-Delay:                       900 Seconds

Volume:                /acct

  Rule
Priority   Rule                       Type             Status
---------  ------------------------   ---------------  ---------------------------------------
    1      docs2das8                  Place            Vol. Scan: Initializing   Migration: 
Initializing
```

```
    2      fm1                       Share Farm       Vol. Scan: Complete   Migration: Complete

bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# policy order-rule placeTest after docs2das8
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Moving the Rule to the Beginning or End

You can use the first or last keyword to move the file-placement rule to the beginning or end of the list:

```
policy order-rule rule1 {first | last}
```

where **first | last** sets the new position for *rule1*.

For example, the following command sequence moves the “docs2das8” rule to the first position:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# policy order-rule docs2das8 first
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Pausing All Rules in a Volume

There are occasions when it is helpful to stop all file migrations, such as times scheduled for backups (see *Restoring a Volume's Files*, on page 4-1 of the *ARX CLI Maintenance Guide*). You can use the policy pause command from priv-exec mode to immediately suspend all policy rules in a given volume:

policy pause namespace ***namespace vol-path***

where

namespace (1-30 characters) is the name of a namespace.

vol-path (1-1024 characters) identifies the volume.

This pauses all of the volume's rules, so that they stop all volume scans and migrations. Clients may change files or directories so that they match a rule and therefore should be migrated; these migrations are queued until policy processing is resumed later. All file-placement rules continue to direct *new* files and directories to their configured storage. (New objects are created at the correct share, so no migrations are necessary.) Pausing has the same effect as an expired duration on a rule's schedule; recall *Setting the Duration (optional)*, on page 12-6.

For example, the following command pauses all rules in the "wwmed~/acct" volume:

```
bstnA(gbl)# end
bstnA# policy pause wwmed /acct
bstnA# ...
```

Resuming all Scans and Migrations in a Volume

You can use the priv-exec form of no policy pause to immediately resume all rule processing in a volume. This restarts rules that were paused from the priv-exec command. This has no effect on scheduled policy pauses, described below.

no policy pause ***namespace vol-path***

For example, the following command sequence un-pauses policy in the "wwmed~/acct" volume:

```
bstnA(gbl)# end
bstnA# no policy pause wwmed /acct
bstnA# ...
```

Pausing on a Schedule

Some installations want to schedule "off hours" for file migrations; for example, you may want to pause all migrations during regularly scheduled backup windows. You can create a schedule (as described earlier) to define the off hours, then pause a volume according to that schedule. This reduces resource contention between clients and the policy engine. From gbl-ns-vol mode, use the policy pause command to pause the current volume's rules on a schedule:

```
policy pause schedule
```

where ***schedule*** (1-64 characters) is the name of a schedule. Use show schedule for a list of configured schedules (see *Showing All Schedules*, on page 12-9).

Each time the schedule fires, the volume stops all volume scans and migrations as described above. The schedule must have a limited duration, or policy never resumes; for instructions on setting a duration on a schedule, see *Setting the Duration (optional)*, on page 12-6.

For example, the following command sequence assigns the "backupWindow" schedule to the "wwmed~/acct" volume:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# policy pause backupWindow
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

Removing the Policy-Pause Schedule

From gbl-ns-vol mode, you can use the no policy pause command to stop pausing policy on a schedule:

```
no policy pause
```

This has no effect on a policy pause command issued from priv-exec mode.

For example, the following command sequence stops all scheduled-policy pauses in the "medarcv~/rcrds" volume:

```
bstnA(gbl)# namespace medarcv volume /rcrds
bstnA(gbl-ns-vol[medarcv~/rcrds])# no policy pause
bstnA(gbl-ns-vol[medarcv~/rcrds])# ...
```

Setting Minimum Free Space For a Share

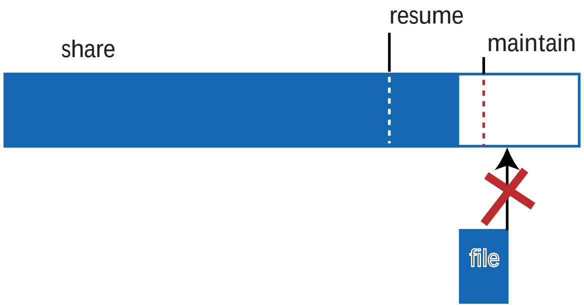
The policy engine has restrictions on the amount of free space it can use on its migration targets. You can use a gbl-ns-vol-shr command to set two free-space parameters for the current share:

- free space to *maintain*,

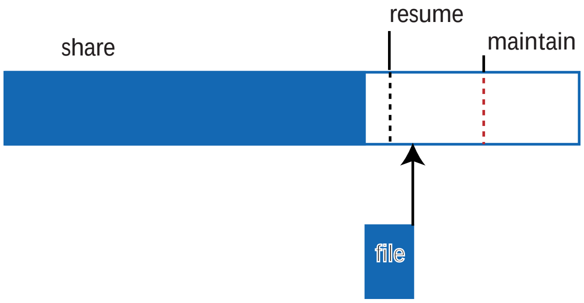
  and, if the share fills to that limit,

- the level of free space that you consider enough to *resume* migrations to the share.

A file-placement rule pauses if one of its files would exceed the *maintain* threshold:



The rule then waits for the share's free space to rise above the *resume* threshold. The policy engine queries the back-end share for its current free space every 15 seconds. If another rule migrates files off of the share until it reaches its *resume* level of free space, the paused rule places the file and resumes its migrations onto the share:



From gbl-ns-vol-shr mode, use the policy freespace command to set the current share's free-space thresholds:

```
policy freespace maintain-space{k|M|G|T} resume-migrate resume-space
```

where

> ***maintain-space*** (any integer) is the lower limit of free space for this share. An ARX rule will not migrate a file onto the share so that it drops below this amount of free space. The default is 1G.

k|M|G|T chooses the unit of measure for the free space: **k**ilobytes, **M**egabytes, **G**igabytes, or **T**erabytes. All values are base-2; e.g., a kilobyte is 1,024 bytes and a megabyte is 1,048,576 bytes. No space is allowed between the number and this unit of measure: **20G** is correct, **20 G** is not.

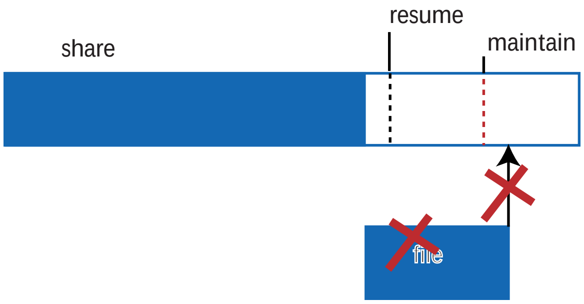
resume-space (as above) is at least as much space as the *maintain-space*, and typically more. If rule X pauses because it would have brought the share below its *maintain-space*, it waits until the share has at least *resume-space* before it starts migrating to this share again. The default is 2G.

For example, the following command sequence sets the "wwmed~/acct~budget" share to maintain at least 5G of free space and to resume as a migration target if its free space rises up to 8G:

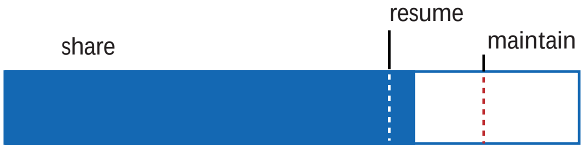
```
bstnA(gbl)# namespace wwmed volume /acct share budget
bstnA(gbl-ns-vol-shr[wwmed~/acct~budget])# policy freespace 5G resume-migrate 8G
bstnA(gbl-ns-vol-shr[wwmed~/acct~budget])# ...
```

Behavior When a File Crosses Both Thresholds

Rules behave differently when the target share is above its *resume* threshold, but a file is big enough to bring the share below its *maintain* threshold:



Ordinarily, a rule should then wait for the above share's free space to rise above the *resume* threshold, but the free space is already above that level. The rule cannot wait for an event that has already occurred, so it skips the file and continues with the next file. It migrates as many files as it can fit on the share while still staying above the *maintain* level, and it skips any files that are too large. The rule continues this process until the share drops below the *resume* limit:



Then the behavior is the same as described above.

Showing Skipped Files and their Sizes

The output of a detailed show policy command indicates the total number of skipped files and their total size. Recall *Focusing on One Share Farm or Rule*, on page 14-43 for instructions on using show policy for a file-placement rule. You can use this information to determine the limitations of the share and/or your placement rules. If the placement rule has verbose reports configured (recall *Generating Verbose Reports*, on page 14-22), the report contains a list of all skipped files.

Setting Free-Space Percentages

You can use the percent keyword to set these free-space limits in terms of total-size percentage. These numbers are a percentage of the back-end share's full volume size:

```
policy freespace percent maintain-pct resume-migrate resume-pct
```

where

maintain-pct (1-100) is the lower percentage of free space for this share. The ARX rule will not migrate a file onto the share so that it drops below this percentage of free space.

resume-pct (1-100) is at least as much space as the *maintain-pct*, and typically much more. If rule X pauses because it would have brought the share below its *maintain-pct*, it waits until the share has at least *resume-pct* of free space before it starts migrating to this share again.

For example, the following command sequence resets the "wwmed~/acct~budget" share to maintain at least 5% of free space and to resume as a migration target if its free space rises up to 7%:

```
bstnA(gbl)# namespace wwmed volume /acct share budget
bstnA(gbl-ns-vol-shr[wwmed~/acct~budget])# policy freespace percent 5 resume-migrate 7
bstnA(gbl-ns-vol-shr[wwmed~/acct~budget])# ...
```

Setting Freespace Limits for an Entire Volume or Namespace

For volumes or namespaces with large numbers of shares, you can use the same command in gbl-ns-vol or gbl-ns mode to set these free-space limits for all shares at once. These commands are macros: they run the above policy freespace command in every share in the current namespace or volume.

```
policy freespace maintain-space resume-migrate resume-space
```

or

```
policy freespace percent maintain resume-migrate resume
```

where all of the options were described above.

For example, the following command sequence sets all shares in the "wwmed~/acct" volume to the same free-space limits:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# policy freespace percent 2 resume-migrate 3
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

These commands do not appear in the global configuration. Only the share-level instances of these commands appear in the global-config file. Use the show global-config namespace command to see these settings for each share: recall *Showing Namespace Configuration*, on page 7-26.

Reverting to the Default Free-Space Thresholds

By default, the policy engine maintains 1G of free space for every share, and resumes migrating to the share if it rises above 2G of free space. From gbl-ns, gbl-ns-vol, or gbl-ns-vol-shr mode, you can use no policy freespace to return to these defaults:

```
no policy freespace
```

For example, the following command sequence sets all shares in the “insur” namespace to the default free-space limits:

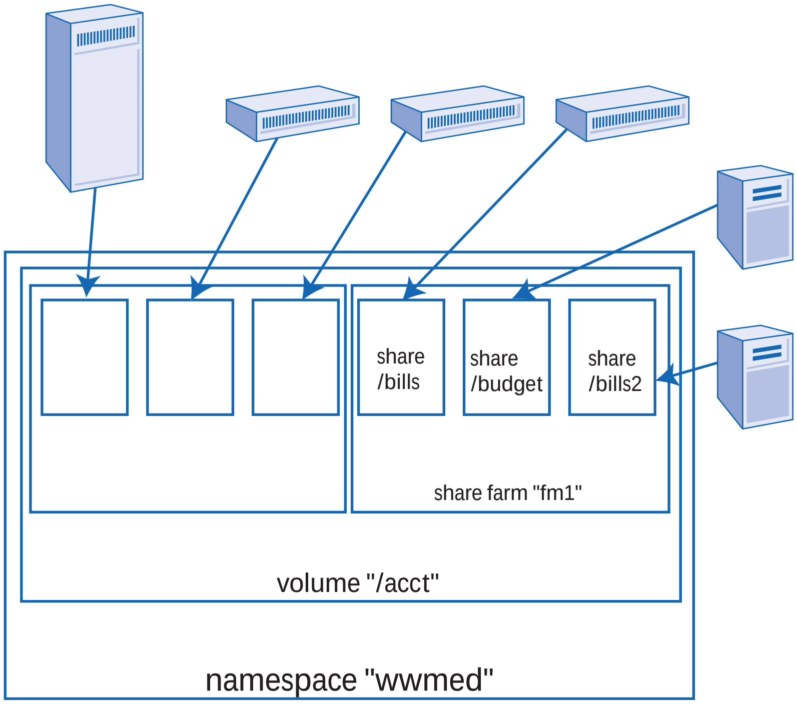
```
bstnA(gbl)# namespace insur
bstnA(gbl-ns[insur])# no policy freespace
bstnA(gbl-ns[insur])# ...
```

15

Grouping Shares in a Share Farm

Overview

As mentioned in earlier chapters, you can group a managed volume’s shares into a *share farm*:



A share farm can expand the storage space for one of the volume’s tiers. The share farm balances this storage space between its shares, using file migration and new-file placement policies. (*New-file placement* is the placement of files that are newly-created by clients.)

Direct volumes, which contain no metadata, do not support share farms. This chapter is relevant to managed volumes only.

You must configure a namespace and at least one managed volume before you configure any share farms. See *Chapter 7, Configuring a Namespace*, and *Chapter 9, Adding a Managed Volume*.

From gbl-ns-vol mode, use the share-farm command to add a share farm to the volume:

share-farm *name*

> where ***name*** (1-1024 characters) is the name you choose for the share farm.

The CLI prompts for confirmation before creating a new share farm; enter **yes** to continue. This puts you into gbl-ns-vol-sfarm mode, where you can identify the shares in the farm.

For example, the following command sequence creates an empty share farm, “fm1:”

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# share-farm fm1

This will create a new share farm.

Create share farm 'fm1'? [yes/no] yes
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# ...
```

CLI Shortcut into gbl-ns-vol-sfarm Mode

You can reach gbl-ns-vol-sfarm mode from gbl mode by typing the name of the namespace, volume, and share farm in the same command. From gbl mode, you can use this syntax:

```
namespace namespace-name volume vol-path share-farm farm-name
```

where

namespace-name identifies the namespace,

vol-path is the volume name, and

farm-name is the share farm.

For example, this command sequence reaches the “wwmed~/acct~fm1” share farm with a single command:

```
bstnA(gbl)# namespace wwmed volume /acct share-farm fm1
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# ...
```

Adding a Share to the Farm

The next step in creating a share farm is to add a share to the farm. The share farm is only useful if it contains two or more shares. Use the share command to add one:

```
share name
```

where ***name*** (1-64 characters) identifies one of the volume’s shares.

For example, the following command sequence adds four shares to the share farm named “fm1:”

```
bstnA(gbl)# namespace wwmed volume /acct share-farm fm1
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# share bills
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# share budget
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# share bills2
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# share it5
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# ...
```

Removing a Share from a Share Farm

Use the no form of the share command to remove a share from the current share farm:

```
no share name
```

where ***name*** (1-64 characters) identifies the share to be removed.

For example, the following command sequence removes the “it5” share from the share farm named “fm1:”

```
bstnA(gbl)# namespace wwmed volume /acct share-farm fm1
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# no share it5
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# ...
```

Constraining New Files

Many installations prefer to retain files on the same back-end shares as their parent directories. This is a recommended practice for a site where backups are made directly from back-end filers instead of through the ARX VIP. By keeping a directory tree together, its easier to find a file’s back-end share for backups and restores. You can skip to the next section for an installation where free-space or bandwidth distribution is a bigger concern.

From gbl-ns-vol-sfarm mode, you can use the constrain-files command to keep any new file in the same share as its parent directory:

constrain-files

This is the opposite of the new-file balance policy described below; if you constrain new files, you cancel the effects of any balance command that is in force.

For example:

```
bstnA(gbl)# namespace ns2 volume /usr share-farm fm4
bstnA(gbl-ns-vol-sfarm[ns2~/usr~fm4])# constrain-files
bstnA(gbl-ns-vol-sfarm[ns2~/usr~fm4])# ...
```

Constraining New Directories

By default, an enabled share farm evenly distributes all new directories amongst its shares, to improve the chances of even free-space distribution. You can instead constrain any new directory to the same share as its parent. The constrain-directories command imposes this constraint.

This only makes sense in a share farm where you constrain new files, too. If new-file balancing is enabled and a client creates several new files in the same directory, that directory must be copied onto each share that receives any of the new files. This is called *striping* the directory. The constrain-directories command cannot prevent this; in this case, it only determines which share has the *master* copy of the directory.

From gbl-ns-vol-sfarm mode, use the constrain-directories command to constrain new directories to the same share as their parent directories:

constrain-directories

For example, the following command sequence causes all new directories in the ns2~/usr~fm4 share farm to remain in the same shares as their parent directories:

```
bstnA(gbl)# namespace ns2 volume /usr share-farm fm4
bstnA(gbl-ns-vol-sfarm[ns2~/usr~fm4])# constrain-directories
bstnA(gbl-ns-vol-sfarm[ns2~/usr~fm4])# ...
```

Constraining Directories Below a Certain Depth

You can apply the directory constraint to any level in the volume's directory tree. For example, consider a volume called /var that gets three new child directories, /usr, /log, and /bin: if you constrain all directories below the first level of this tree, the share farm can distribute those directories to any share with available space. The constraint applies to directories below that first level: any directory created inside /usr, /log, or /var is constrained to the same share as its parent directory.

To constrain directories below a certain depth, add the optional below-depth argument to the constrain-directories command:

constrain-directories below *depth*

> where ***depth*** (0-100) is the highest level in the volume's directory tree where balance is still enforced. The directory trees below that level are constrained to the same share as their parent directories. This defaults to 0, which constrains all directories in the volume.

For example, the following command sequence causes all new directories in the ns2~/usr~fm4 share farm to be constrained to their parent directories when the new directory is below the first subdirectory level.

```
bstnA(gbl)# namespace ns2 volume /usr share-farm fm4
bstnA(gbl-ns-vol-sfarm[ns2~/usr~fm4])# constrain-directories below 1
bstnA(gbl-ns-vol-sfarm[ns2~/usr~fm4])# ...
```

Given a depth of 'below 1', if the ns2~/usr~fm4 share farm has shares s1, s2, and s3 with the same free space and capacity, and you have created root directories /a, /b, and /c, these directories are spread evenly across shares s1, s2, and s3 one share at a time. Any directory created in /a, /b, or /c is placed on the same share as its parent directory.

Not Constraining Directories

Use no constrain-directories to remove directory placement restrictions and have new directories distributed as directed by one of the balance commands.

no constrain-directories

For example:

```
bstnA(gbl)# namespace ns2 volume /var share-farm fm2
bstnA(gbl-ns-vol-sfarm[ns2~/var~fm2])# no constrain-directories
bstnA(gbl-ns-vol-sfarm[ns2~/var~fm2])# ...
```

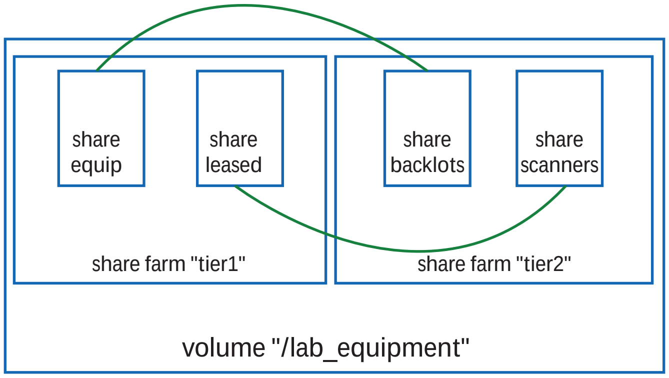
Client Restrictions When a Share is Unavailable

When new-file or new-directory distribution is constrained, a user may get an error if one share in the farm is unavailable (offline or otherwise unusable). The share farm is constrained to creating the new file or directory on the same share as its parent, so the user gets a create error if that share goes offline.

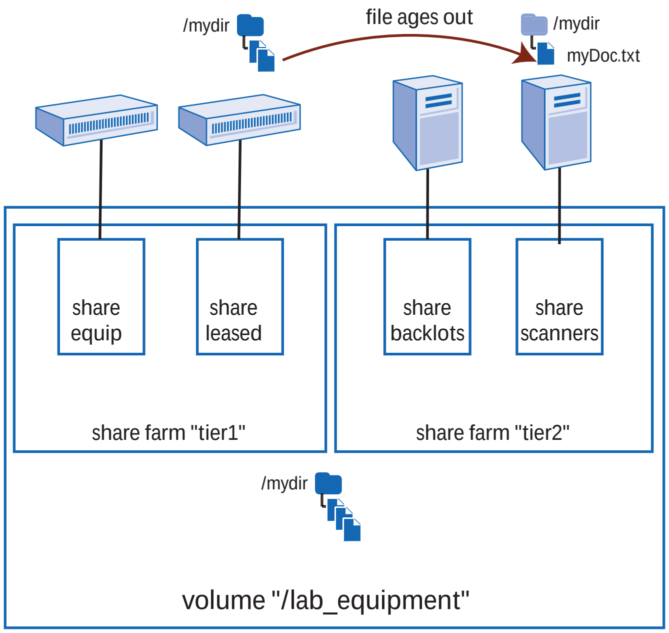
Constraints in a Tiered Volume: Share Farm Mirroring

A volume with tiered storage may contain multiple share farms, one for each tier. Tier 1 storage is typically backed by the most reliable filers, so we recommend keeping all master directories on Tier 1 shares (recall *Best Practice for Tiering: Promote all Directories In Tier 1*, on page 14-11). If you follow this best practice, all master directories reside on the Tier 1 share farm and none of them reside on any other share farm. The constraints in this section are designed to keep new files and directories on the same share as their parent; that is, on the same share as the parent's master directory. However, a file that migrates to the Tier 2 share farm cannot go to the share with the master directory because all master directories reside in the Tier 1 farm. To determine the correct share for such a file, the managed volume uses *share-farm mirroring*.

Share-farm mirroring means mapping shares between two or more farms. Specifically, each share in one share farm is associated with a corresponding share in the other share farm(s). Consider the two share farms in "medarcv~/lab_equipment" from some of the examples above. This illustration connects the mirrored shares with curved lines:



The master directories in the "equip" share are striped in the "backlots" share, and the "leased" share is mirrored by the "scanners" share. Suppose the "tier 2" share farm has file constraints and a file named “mydir\myDoc.txt” has aged out. If the master directory for “mydir” is on the “leased” share, the share farm sends the file to its mirror share, “scanners:”



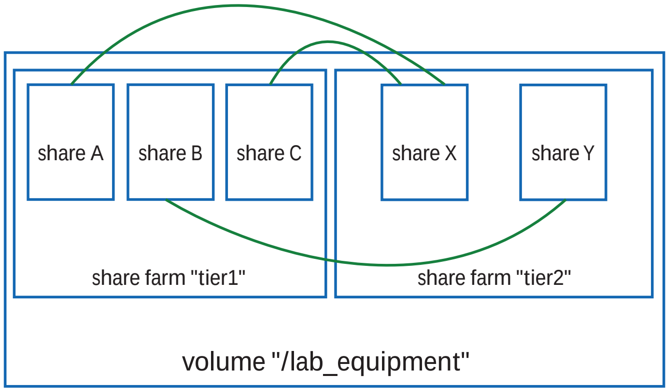
This causes the directory trees to mirror each other. The master for “mydir” grows on the “leased” share while the “mydir” stripe directory grows on the “scanners” share. The “leased” share holds all of mydir’s Tier 1 files, and the “scanners” share holds all of mydir’s files that have been demoted to Tier 2.

Keeping the Same Number of Shares in Each Share Farm

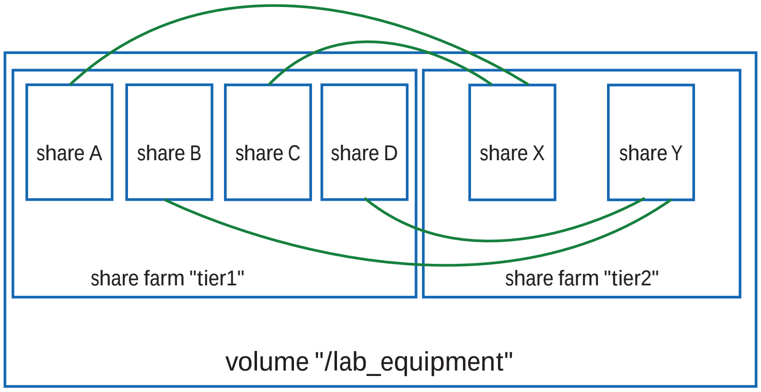
Share-farm mirroring is designed for volumes where all tiers are share farms with the same number of shares. If the first tier is a single share and lower tiers are share farms, the policy engine cannot implement share-farm mirroring at all. The file or directory constraints would therefore force all files onto the first-configured share in each share farm. The extra share(s) in each share farm remain empty.

If a Tier 1 farm has more shares than the Tier 2 or Tier 3 farm, some shares on the lower-tier farm must take the stripe directories from the extra shares in the Tier 1 farm. For example, if the Tier 1 farm has 3 shares and the Tier

2 farm has 2 shares, the first share in the Tier 2 farm (share X from the figure) holds the stripe directories for the first and third shares (shares A and C) in the Tier 1 farm:

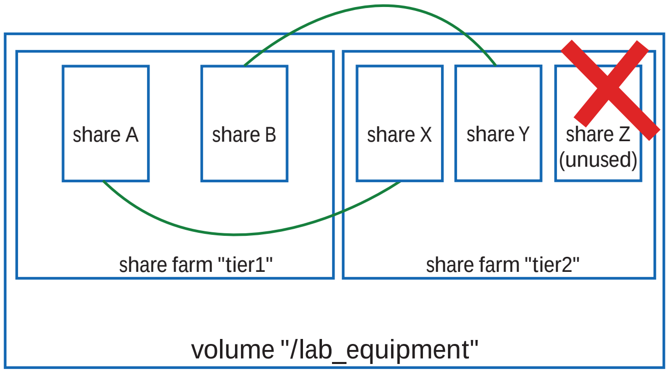


If the imbalance is 4 shares to 2 shares, both shares in the Tier 2 farm must host the stripes for 2 shares each:



The next share in Tier 1 would be reflected in Tier 2's share X, and then the next share would be reflected in share Y, and so on.

If the reverse imbalance is true, where the Tier 2 share farm has more shares than the Tier 1 farm, the policy engine does not use the extra shares in the Tier 2 farm for any files. The extra shares remain empty due to the constraints:



Balancing New Files Instead of Constraining Them

Use the no form of the constrain-files command to balance new files in the current share farm. With this (default) setting, the ARX uses the new-file balancing algorithm described in the next section:

```
no constrain-files
```

This distributes new files evenly, without regard to keeping directories together on the same back-end share. This also allows share farms of unbalanced sizes to co-exist in a tiered volume.

For example, this command sequence disables file constraints in the “fm4” share farm:

```
bstnA(gbl)# namespace ns2 volume /usr share-farm fm4
bstnA(gbl-ns-vol-sfarm[ns2~/usr~fm4])# no constrain-files
bstnA(gbl-ns-vol-sfarm[ns2~/usr~fm4])# ...
```

An even distribution of new files can complicate some backup and restore scenarios. If your filers support snapshots and/or checkpoints, you can use *file tracking* to locate the correct backup tapes for a given file (see *Chapter 3, Tracking Files on Your Back-End Storage* in the *ARX CLI Maintenance Guide*).

Balancing New Files Based on Free Space

Some installations do not require file or directory constraints on back-end shares. These are installations that perform backups and restores through the ARX VIP (on the client side) and/or use ARX snapshots (see *Chapter 2, Configuring Volume Snapshots* in the *ARX CLI Maintenance Guide*).

A share farm can balance all new files evenly amongst its shares. A *new file* is a file created by a client or a file that migrates to the share farm from another share in the same volume. For example, consider a share farm with shares s1 and s2, where both shares are equally weighted: the first new file goes to s1, the second goes to s2, the third goes to s1, and so on. With file and directory constraints disabled, this is the default algorithm for distributing new files.

You can configure the share farm to assign new files based on the current free space available on each share. (The ARX gets its free-space numbers from each share every 15 seconds.) This algorithm assigns new files based on the relative free space at each share: for example, if s1 has two gigabytes of free space and s2 has one gigabyte of free space, s1 is assigned twice as many new files as s2.

◆ **Note**

This balancing of new files cannot be intelligent about the size of files created by clients; a file's size is always zero at the moment it is created. After the file is written and its size is established, an auto-migrate *rule (described later) can migrate the file to an emptier share as needed.*

New files from shares outside the share farm, migrated to the share farm by a file-placement rule, have a size at migration time. The share farm can therefore place them according to capacity.

To balance new-file distribution based on free space, use the balance capacity command from gbl-ns-vol-sfarm mode:

```
balance capacity
```

For example, the following command sequence distributes new files in the 'fm1' share farm based on relative free space (capacity) at each share:

```
bstnA(gbl)# namespace wwmed volume /acct share-farm fm1
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# balance capacity
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# ...
```

Based on Latency (Bandwidth)

The NSM continuously updates its measure of the average *latency* (round-trip packet time) between its ports and each share. A low latency for a share indicates high bandwidth at the share. You can use the balance command to distribute new files based on latency measures instead of free-space measures. For example, consider a share farm where one share, s1, has an average latency that is three times faster than the latency to s2: this algorithm would send three times as many files to s1 as s2. The share with the best latency gets the most new files.

To distribute more new files to shares with lower latency (and therefore more bandwidth), use the balance latency command:

```
balance latency
```

For example, the following command sequence configures the ‘medFm’ share farm to distribute its new files based on the current latency at each share:

```
bstnA(gbl)# namespace medarcv volume /rcrds share-farm medFm
bstnA(gbl-ns-vol-sfarm[medarcv~/rcrds~medFm])# balance latency
bstnA(gbl-ns-vol-sfarm[medarcv~/rcrds~medFm])# ...
```

Based on Administrative Weights

You can also choose to use the weighted-round-robin method of distributing new files, using the weights that you set when you add each share to the share farm. This is the default new-file-placement policy for a new share farm. The number of new files assigned to each share is based on the weight of the share, relative to all the other share weights: for example, if s1 has a weight of 15 and s2 has a weight of 60, s2 gets four times as many new files as s1. A share weight can be any number between 0 and 100.

The default weight is 1. From gbl-ns-vol-sfarm mode, use the weight clause with the share command to set the share’s weight:

```
share name weight weight
```

where

name (1-64 characters) identifies the share, and

weight (0-100) is the weight of the share, relative to the weights you set for other shares in the same farm. A 0 (zero) makes the share ineligible for new files.

To place the new files on the shares based on their relative weights, use the balance round-robin command from the same mode:

```
balance round-robin
```

For example, the following command sequence causes the share farm to assign twice as many files to the ‘back1’ share as the ‘back2’ share:

```
prtlndA(gbl)# namespace nemed volume /acctShdw share-farm farm1
prtlndA(gbl-ns-vol-sfarm[nemed~/acctShdw~farm1])# share back1 weight 20
prtlndA(gbl-ns-vol-sfarm[nemed~/acctShdw~farm1])# share back2 weight 10
prtlndA(gbl-ns-vol-sfarm[nemed~/acctShdw~farm1])# balance round-robin
prtlndA(gbl-ns-vol-sfarm[nemed~/acctShdw~farm1])# ...
```

The Share Farm Maintains Each Share's Minimum Free Space

Each ARX share has a minimum amount of free space that the policy engine attempts to maintain. You can use the policy freespace command to set this minimum, as described in *Setting Minimum Free Space For a Share*, on page 14-52. The policy engine never migrates a file to any share if the file would reduce the share's free space below this minimum, but clients may exceed the minimum by adding to files that are already on the share. If the share farm is balancing new files, it only places new files on shares that are above their minimum free space. The share farm does not overload any shares that are below their policy freespace thresholds.

◆ **Note**

Conversely, this does not apply to a share farm with file or directory constraints. That is, if a file must migrate to a share due to constraints, the policy engine ignores the share's free-space threshold.

New-File Placement When All Shares Reach the Free Space Threshold

If all shares fill up to their policy freespace limits, the share farm distributes each new file to the share with its master directory. For example, if a client created "bigDir/myList.xls" in a share farm, and the master directory for "bigDir" was on share X, the new "myList.xsl" file is created on share X. If the master directory is not on the share farm, the file goes to the first-configured share in the farm.

This situation can cause a share to exceed its policy freespace threshold. The policy engine sends SNMP traps whenever any share in the farm approaches this threshold, so the free space issue should be addressed before the situation occurs.

Setting a Free-Space Threshold for all Shares in the Farm at Once

For share farms with large numbers of shares, you can use the policy freespace command in gbl-ns-vol-sfarm mode to set these free-space limits for all the shares in the farm. This commands is a macro: it runs the policy freespace command in every share in the current share farm. It is similar to the commands in gbl-ns and gbl-ns-vol mode, described in *Setting Freespace Limits for an Entire Volume or Namespace*, on page 14-54. The command syntax is described there.

For example, the following command sequence sets all shares in the "fm1" farm to maintain at least 2G of free space, and resume as a migration target if the free space rises to 4G:

```
bstnA(gbl)# namespace wwmed volume /acct share-farm fm1
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# policy freespace 2 resume-migrate 4
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# ...
```

This command does not appear in the global configuration. The CLI invokes the command in each share, and only those share-level instances of the command appear in the global-config file. Use the show global-config namespace command to see these settings for each share: recall *Showing Namespace Configuration*, on page 7-26.

Auto Migrating Existing Files

This section only applies to a share farm that balances new files based on share capacity (as described in *Balancing New Files Based on Free Space*, on page 15-11). Auto migration also cannot function in a share farm that uses file or directory constraints. Skip to the next section if your share farm uses file or directory constraints, or if it uses a different setting for the new-file balance command.

You can configure an *auto-migrate* policy to migrate files off of a share that is low on free space. These are existing files as opposed to the empty new files discussed above. The files migrate to shares that are *not* low on free space, if there are any such shares in the same share farm.

A share is considered low on free space if it drops below its *maintain* threshold, which you can set with the policy freespace command shown above. For a detailed discussion of this threshold and command, refer back to *Setting Minimum Free Space For a Share*, on page 14-52. If clients fill a share to this level or if a rule approaches it, auto migration can migrate files off of the share. New files are not allowed on the share during an auto migration. The auto migration continues until the share's free space rises above its *resume* threshold; that is, until other rules can resume using the share as a migration target.

From gbl-ns-vol-sfarm, use the auto-migrate command to allow auto-migration in the current share farm:

```
auto-migrate
```

For example, the auto-migrate command in the following command sequence ensures that if any share the 'fm1' share farm drops below its minimum free space, the share farm migrates some of its files to emptier shares:

```
bstnA(gbl)# namespace wwmed volume /acct share-farm fm1
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# auto-migrate
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# ...
```

Recommended for File-Placement Targets

When a share farm is a file-placement target, the first configured share in the farm is the default share for placed files. Most files are placed on the same share as their parent directory, but a file defaults to the first share if its parent's master directory is outside the share farm. The first share in the farm can therefore take a heavier file burden over time. An auto-migrate directive migrates files off of this share if it drops below its free-space threshold.

Disabling Auto Migration

The auto-migrate policy is disabled by default. Use the no auto-migrate command to return to this default:

no auto-migrate

For example, the no auto-migrate command in the following command sequence disables automatic file migration in the ‘medFm’ share farm.

```
bstnA(gbl)# namespace medarcv volume /rcrds share-farm medFm
bstnA(gbl-ns-vol-sfarm[medarcv~/rcrds~medFm])# no auto-migrate
bstnA(gbl-ns-vol-sfarm[medarcv~/rcrds~medFm])# ...
```

Enabling All Share-Farm Rules

The final step in configuring a share farm is to enable its rules. By default, the share-farm and its rules are disabled and ignored by policy software. From gbl-ns-vol-sfarm mode, use the enable command to enable the share farm:

```
enable
```

For example, the following command sequence enables the ‘fm1’ share farm:

```
bstnA(gbl)# namespace wwmed volume /acct share-farm fm1
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# enable
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~fm1])# ...
```

Stopping All Share-Farm Rules

You can stop all auto migrations and/or new-file balancing on a share farm by disabling it. This reverts all shares to standard behavior; no auto migrations as free space gets low on a share, and any new file or directory is created on the same share as its parent. To do this, use the no enable command from gbl-ns-vol-sfarm mode.

```
no enable
```

For example, the following command sequence stops all rules in the ‘fm2’ share farm:

```
bstnA(gbl)# namespace wwmed volume /acct share-farm tst
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~tst])# no enable
bstnA(gbl-ns-vol-sfarm[wwmed~/acct~tst])# ...
```

Removing a Share Farm

You cannot remove a share farm if any rules use it as a source or target. Remove all such rules before removing the share farm; use show policy to find them, as instructed in *Showing All Rules*, on page 14-34.

From gbl-ns-vol mode, use the no share-farm command to remove a share farm:

```
no share-farm name
```

where ***name*** (1-1024 characters) identifies the share farm to be removed.

This does not affect client access to the farm's shares.

For example, the following command sequence removes the "san14luns4-8" share farm from the "/acct" volume:

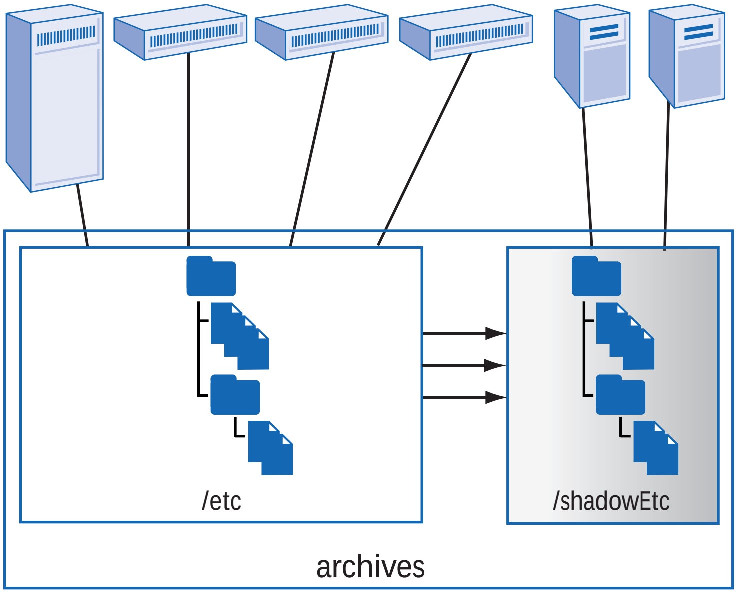
```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# no share-farm san14luns4-8
bstnA(gbl-ns-vol[wwmed~/acct])# ...
```

16

Shadowing a Volume

Overview

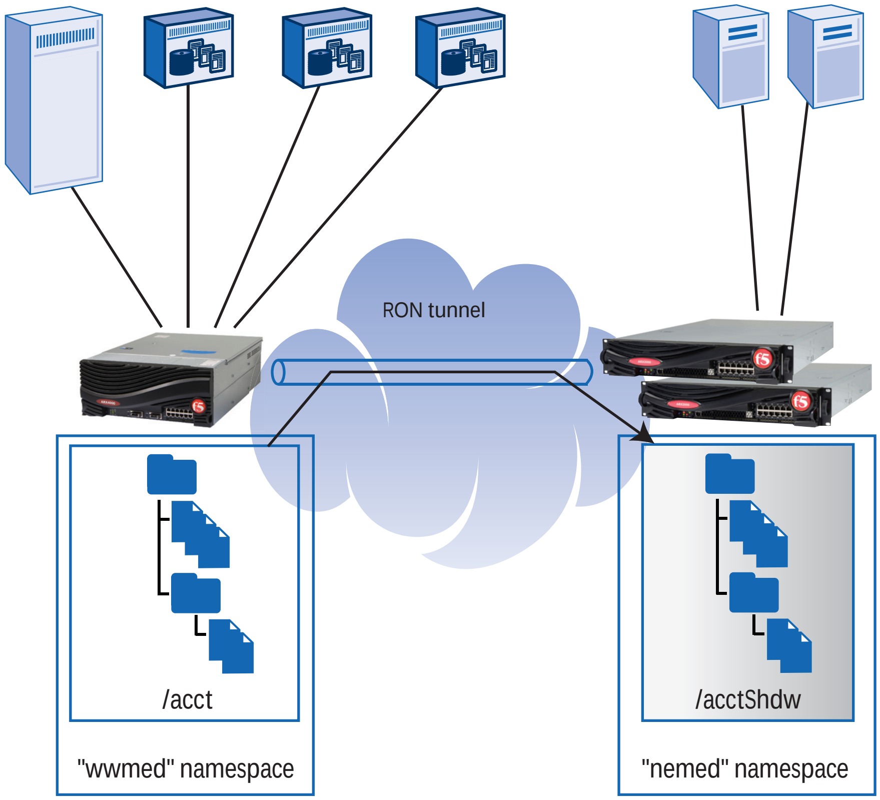
A *shadow volume* keeps regularly-updated copies of a managed volume's files and directories. The shadow volume can be on the same switch as its source volume (as shown below), or it can be hosted on another switch in the same RON (shown on the next page).



Shadow volumes have applications for backing up client data or making remote copies of the data, perhaps for processing at the remote site. In either application, you can funnel several source volumes into the same shadow volume.

The source and shadow volume can have a completely different configurations, as long as the shadow volume has at least as much storage capacity as the source. For example, the shadow volume can be backed by a different number of filers, perhaps with different storage policies. You can customize the volume configurations according to the needs of each site.

The examples in this chapter configure a source volume on a single ARX-4000 and its shadow volume on a redundant pair. The redundant pair is two ARX-2000 switches:



The switch with the source volume is called the *source switch* and any switch with a shadow volume is called a *target switch*.

Before You Begin

Shadow volumes are commonly deployed on separate switches from the source volume, as pictured above. Before you configure the shadow volume on a target switch, you must first

1. make a RON tunnel from the source switch to the target switch (see *Chapter 6, Joining a RON*, in the *ARX® CLI Network-Management Guide*), and
2. add a namespace and source volume at the *source* switch (see *Chapter 7, Configuring a Namespace* and *Chapter 9, Adding a Managed Volume*).
3. add a namespace at the *target* switch.

You then perform the procedures in this chapter: you start by adding a shadow volume to the target switch, then you configure a shadow-copy rule at the source switch.

Adding a Shadow Volume (Target Switch)

The first step in shadowing a volume is configuring a second managed volume as a shadow. A shadow volume is different from a standard managed volume in that it can only contain replicas of source-volume files, and only the policy engine can write to it.

A shadow volume starts as managed volume (see *Adding a Volume*, on page 7-22). To create the shadow volume on a different switch from the source volume, log into the target switch's CLI or GUI and add a new volume. Once the volume is created and you are in gbl-ns-vol mode, use the shadow command to change the managed volume into a shadow volume:

```
shadow
```

The CLI presents a prompt to warn that all extraneous files will be removed from the shadow volume's shares; you must answer **yes** for the volume to become a shadow volume. During the first shadow-copy operation, the rule replaces any files that are different from their source-file counterparts, and then it deletes all files that are *not* among the source files.

Choose a shadow volume with at least as much storage capacity as its source volume(s).

◆ **Note**

The volume must be disabled when you change it into a shadow volume.

For example, the following command sequence creates a shadow volume to be used for the "wwmed~/acct" volume later. Note that this CLI session occurs on a different switch from the one where the "/acct" volume was created; this volume is created on “prtlndA,” a chassis in the same RON as the “bstnA” chassis. The namespace is also different; “nemed” instead of “wwmed.” The two volumes reside on separate switches and namespaces for added redundancy.

```
prtlndA(gbl)# namespace nemed
prtlndA(gbl-ns[nemed])# volume /acctShdw
This will create a new volume.

Create volume '/acctShdw'? [yes/no] yes
prtlndA(gbl-ns-vol[nemed~/acctShdw])# shadow

This will cause the all shares in the volume to be erased.

Are you sure you want to remove all data? [yes/no] yes
prtlndA(gbl-ns-vol[nemed~/acctShdw])#
```

Compatible Software Releases

The source and target switches can typically run different software releases, to facilitate staged software upgrades in you network.

Allowing Modifications

The shadow-copy rule will be modifying the metadata in the target volume, so the target volume must permit metadata modifications. This does not affect clients, who will have read-only access to the volume. It only applies to the shadow-copy rule itself. Use the modify command to enable file modifications by the rule. (The modify command was discussed in an earlier chapter; recall *Allowing the Volume to Modify on Import*, on page 9-6.)

For example, the following command sequence permits modifications in the /acctShdw volume:

```
prtlndA(gbl)# namespace nemed volume /acctShdw
prtlndA(gbl-ns-vol[nemed~/acctShdw])# modify

Automatically re-enable volume modify mode during NSCK rebuild? [yes/no] yes
prtlndA(gbl-ns-vol[nemed~/acctShdw])# ...
```

Adding a Share

A share in a shadow volume is configured the same way as any other managed-volume share, but it is used differently. The shadow-copy rule will replace all files on a shadow-volume share when it first runs; a share in a shadow volume is used exclusively for file replicas from the source volume. To guard against a single point of failure, we strongly recommend that you use shares from filers *other* than those used for the source volume. For maximum protection, use shares from filers at another ARX in the RON.

For example, the following command sequence adds two shares to the /acctShdw volume:

```
prtlndA(gbl)# namespace nemed volume /acctShdw share back1
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back1])# filer das-p1 path /lhome/exports/BU
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back1])# enable
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back1])# exit
prtlndA(gbl-ns-vol[nemed~/acctShdw])# share back2
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back2])# filer das-p2 path /export/home/bkup
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back2])# enable
prtlndA(gbl-ns-vol-shr[nemed~/acctShdw~back2])# exit
prtlndA(gbl-ns-vol[nemed~/acctShdw])# ...
```

CIFS Subshares are Not Supported

Some filers do not allow the ARX to remove directories behind their CIFS subshares, which is necessary if the corresponding directories are removed at the source volume. A shadow volume therefore does not support CIFS subshares (described in *Supporting Subshares and their ACLs*, on page 9-20).

Turning Off Shadowing

You can turn off shadowing to convert a shadow volume back to a managed volume. This makes the volume ineligible for shadow copies and opens it up for client writes; all the shadow-copied files become fully accessible. From gbl-ns-vol mode, use the no shadow command to disable shadowing for the current volume:

```
no shadow
```

You must disable the shadow volume before you turn off shadowing.

A warning message appears, indicating that this erases any unpublished files and shadow databases. This means that files copied to the shadow volume to a hidden staging area, not yet “published” in their final directories, will be removed rather than moved into the final directory. Enter **yes** to continue.

For example, the following command sequence turns off shadowing for the “/buTest” volume:

```
bstnA(gbl)# namespace archives volume /buTest
bstnA(gbl-ns-vol[archives~/buTest])# no enable
bstnA(gbl-ns-vol[archives~/buTest])# no shadow

This will cause any un-published files and shadow databases to be
erased.

Are you sure you want to proceed? [yes/no] yes
bstnA(gbl-ns-vol[archives~/buTest])# ...
```

Specifying a Fileset to Copy (Source Switch)

The next step in shadowing a volume is to choose a fileset to be “shadowed.” This occurs at the source volume, on the source switch. The fileset can include all files in the volume or a smaller set of files based on file names and/or file-access times. You can create complex filesets by intersecting two or more filesets (for example, choosing the *.mp3 files accessed in the last week) and/or joining them in a union (for example, choosing all .mp3 files *plus* all files of *all* types that were accessed in the last week). Refer back to *Chapter 13, Grouping Files Into Filesets*, for a full range of options in creating filesets.

For example, the following command sequence occurs on “bstnA,” the switch with the source volume. These commands create an all-inclusive fileset called “worthSaving:”

```
bstnA(gbl)# policy-filename-fileset worthSaving
This will create a new policy object.

Create object 'worthSaving'? [yes/no] yes
bstnA(gbl-fs-name[worthSaving])# recurse
bstnA(gbl-fs-name[worthSaving])# exit
bstnA(gbl)# ...
```

Shadow Copies Are Not Cumulative

Note that if the fileset changes, the shadow changes with it. For example, consider an age-based fileset which changes based on client access. Files that are in the fileset one day may not be there the next. The shadow copy does not keep old files that are no longer in the source fileset: the changes to the shadow copy are not cumulative.

Configuring a Shadow-Copy Rule (Source Switch)

The final step in volume shadowing is to create a shadow-copy rule. A *shadow-copy rule* replicates the fileset in the source volume over to the shadow volume; if a file changes later, that file is replicated again. If a file is deleted, then its copy is also deleted. From gbl-ns-vol mode (in the *source* volume), use the shadow-copy-rule command to create a shadow-copy rule:

shadow-copy-rule *name*

where ***name*** (1-1024 characters) is a name you choose for this rule.

The CLI prompts you for confirmation before creating the new rule; enter **yes** to continue. This places you into gbl-ns-vol-shdwcp mode, where you must identify the source fileset and the target shadow volume, make a schedule for the rule, then enable the rule. There are also several optional commands in this mode.

For example, the following command sequence creates a shadow-copy rule, “SVrule,” for the “/acct” volume. Like the previous example, this occurs on the “bstnA” switch:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule

This will create a new policy object.

Create object 'SVrule'? [yes/no] yes
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Identifying the Source Fileset

The next step in configuring a shadow-copy rule is to identify the source fileset. The shadow-copy rule keeps replicating all files from the source fileset onto the shadow volume. From gbl-ns-vol-shdwcp mode, use the from fileset command to specify a source fileset:

from fileset *fileset-name* [match {files | all}]

where

fileset-name (1-1024 characters) identifies the source fileset.

match {files | all} (optional) determines whether to use the fileset to select files only or files and directories. If you omit this, the fileset matches files only.

You can only use one source fileset. If you want to use additional filesets as sources, create a union fileset (see *Joining Filesets*, on page 13-18). You can re-issue the from command to change from one source fileset to another.

For example, the following command set selects the “worthSaving” fileset as a source for the shadow copy:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# from fileset worthSaving
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Choosing a Shadow-Volume Target

The next step in configuring a shadow-copy rule is to choose a target volume for the shadow copy. A target must be a shadow volume in the same RON. From gbl-ns-vol-shdwcp mode, use the target command to identify a shadow volume:

target [[hostname *host*] namespace *name*] volume *shadow*

where

hostname *host* (optional, 1-128 characters) identifies the target switch that hosts the shadow volume. If you omit this, the source switch is assumed. The switch must be on the same RON as the source switch; use the show ron command for a list of all switches on the RON (see *Showing the RON Configuration*, on page 6-6 of the *ARX® CLI Network-Management Guide*).

namespace *name* (optional, 1-30 characters) identifies the namespace containing the shadow volume. If you omit this, it defaults to the current namespace.

shadow (1-256 characters) identifies the shadow volume.

For example, the following command sequence selects "/acctShdw" as a shadow volume for "SVrule:"

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# show ron

Switch Name             HA Peer Switch                                  Uptime
Status                  UUID                                   Management Addr
------------------------------------------------------------------------------
bstnA                   (None)                                0 days, 03:27:36
ONLINE                  d9bdece8-9866-11d8-91e3-f48e42637d58          10.1.1.7

canbyA                  (None)                                0 days, 03:29:56
ONLINE                  64a6417e-cc3d-11df-80ca-a73fbeb72ef8       10.1.33.105

newptA                  (None)                                0 days, 03:16:07
ONLINE                  cf251849-826d-01a8-9110-8dtu78fca5b2       10.1.117.74

provA                   (None)                                0 days, 03:26:27
ONLINE                  db922942-876f-11d8-9110-8dtu78fc8329        10.1.38.19

prtlndA                 prtlndB                               0 days, 03:29:14
ONLINE                  876616f6-79ac-11d8-946f-958fcb4e6e35        10.1.23.11

prtlndB                 prtlndA                               0 days, 03:28:08
ONLINE                  64dcab94-a2b6-11d8-9d25-bf2c991c83f9        10.1.23.12

stkbrgA                 (None)                                0 days, 03:28:57
ONLINE                  8fa98111-55ec-d1c8-9380-8dtu78fab47d     192.168.66.62

stoweA                  (None)                                0 days, 03:29:58
ONLINE                  05d5a0fa-f2fb-11df-8daf-af50d57e388e        10.1.14.76


bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# target hostname prtlndA namespace nemed volume
/acctShdw
```

```
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Using a Path in the Shadow Volume

You may want to consolidate several source volumes into a single shadow volume, for ease of backups. Each source volume can store its files in a separate directory in the shadow volume. Use the optional path argument to create such a directory inside the volume:

target [[hostname *host*] namespace *name*] volume *shadow* path *path*

where

host, ***name***, and ***shadow*** are described above, and

path (optional, 1-1024 characters) is the path where the copies will be stored. This is a path from the root of the volume.

For example, the following command sequence selects two directories as shadow-copy targets:

```
prtlndA(gbl)# namespace testns
prtlndA(gbl-ns[testns])# volume /users
prtlndA(gbl-ns-vol[testns~/users])# shadow-copy-rule bkup
prtlndA(gbl-ns-vol-shdwcp[testns~/users~bkup])# target volume /shdw path /users
prtlndA(gbl-ns-vol-shdwcp[testns~/users~bkup])# exit
prtlndA(gbl-ns[testns])# volume /admin
prtlndA(gbl-ns-vol[testns~/admin])# shadow-copy-rule bkAdm
prtlndA(gbl-ns-vol-shdwcp[testns~/admin~bkAdm])# target volume /shdw path /admin
prtlndA(gbl-ns-vol-shdwcp[testns~/admin~bkAdm])# ...
```

Removing the Shadow-Volume Target

From gbl-ns-vol-shdwcp mode, use no target to remove a shadow-volume target:

no target [[hostname *host*] namespace *name*] volume *shadow* [path *path*]

where

host (1-128 characters) identifies the target switch that hosts the shadow volume. The default is the local (source) switch.

name (1-30 characters) identifies the namespace containing the shadow volume. The default is the current namespace.

shadow (1-256 characters) identifies the shadow volume.

path (optional, 1-1024 characters) is a specific directory to stop targeting.

If you remove the target, the shadow-copy rule is effectively disabled.

For example, the following command sequence removes a target volume and path from the "bkup" rule:

```
prtlndA(gbl)# namespace testns volume /users
prtlndA(gbl-ns-vol[testns~/users])# shadow-copy-rule bkup
prtlndA(gbl-ns-vol-shdwcp[testns~/users~bkup])# no target volume /shdw path /users
prtlndA(gbl-ns-vol-shdwcp[testns~/users~bkup])# ...
```

Applying a Schedule

A shadow-copy rule requires a schedule. Use the gbl schedule command to create one; refer back to *Chapter 12, Creating a Policy Schedule* for details.

◆ **Note**

You cannot use a schedule with a fixed duration *(see Setting the Duration (optional), on page 12-6). If a duration was too short for the shadow copy to finish, the shadow copy would fail.*

To apply a schedule, go to gbl-ns-vol-shdwcp mode and use the schedule command:

```
schedule name
```

where ***name*** (1-64 characters) identifies the schedule. Use the show schedule command to list all schedules.

For example, the following command sequence applies an hourly schedule to the shadow-copy rule, “SVrule:”

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# schedule hourly
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Rule-Configuration Changes in a Scheduled Rule

After you apply a schedule to an enabled rule, other configuration changes wait until the next time the schedule fires. For example, if you change the target share on a shadow-copy rule with a schedule, the rule continues to use the original target share until the next scheduled run. This only applies to an enabled rule, one that is already running; the method for enabling a rule is described later in the chapter.

You can make a rule change effective immediately by disabling and then re-enabling the rule.

Configuring Progress Reports

The next step in configuring a shadow-copy rule, optional but strongly recommended, is to arrange for progress reports. Progress reports show all the milestones and results of a shadow-copy execution. The policy engine generates a report each time the schedule fires and invokes the rule.

By default, no reports are generated. From gbl-ns-vol-shdwcp mode, use the report command to generate shadow-copy reports for the current rule:

```
report prefix
```

where ***prefix*** (1-1024 characters) is the prefix to be used for the rule’s reports. Each report has a unique name in the following format: *prefixYearMonthDayHourMinute*.rpt (for example, home_backup200403031200.rpt for a report with the “home_backup” prefix).

Use the show reports command for a list of all reports, or show reports type SCp for a list of shadow-copy reports. Use show reports *report-name* to read a report, show reports status *report-name* for a one-line summary, grep to search through the report, or tail to tail a report as it is being written.

For example, the following command sequence enables reporting for the rule, “SVrule:”

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# report SVetc
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Generating Verbose Reports

Shadow-copy reports are terse by default. To make them verbose, use the optional verbose flag at the end of the report command:

report *prefix* verbose

where ***prefix*** is explained above.

For example, the following command resets “SVrule” to produce verbose reports:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# report SVetc verbose
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Including Identical Files in Reports

If files are identical on both the source and shadow volumes, the rule does not transfer the file. By default, identical files are omitted from the shadow-copy reports. Use the optional list-identical flag to include these files in the report:

report *prefix* [verbose] list-identical

where ***prefix*** and **[verbose]** are explained above.

For example, the following command sets the “insurSV” rule to produce verbose reports and include identical files:

```
bstnA(gbl)# namespace insur volume /claims
bstnA(gbl-ns-vol[insur~/claims])# shadow-copy-rule insurSV
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# report insurSV verbose list-identical
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# ...
```

Deleting Empty Reports

By default, a shadow-copy rule creates a report every time it runs, even in cases where no files are copied between volumes. To remove any empty-report files created this way, use the optional delete-empty flag at the end of the report command:

report *prefix* [verbose] [list-identical] delete-empty

where ***prefix***, **[verbose]**, and **[list-identical]** are explained above.

For example, the following command resets “SVrule” to delete any empty shadow-copy reports:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# report SVetc delete-empty
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Disabling Reports

From gbl-ns-vol-shdwcp mode, use no report to stop generating shadow-copy reports for the current rule:

```
no report
```

For example, the following command sequence disables reporting for the rule, “buHome:”

```
bstnA(gbl)# namespace archives volume /home
bstnA(gbl-ns-vol[archives~/home])# shadow-copy-rule buHome
bstnA(gbl-ns-vol-shdwcp[archives~/home~buHome])# no report
bstnA(gbl-ns-vol-shdwcp[archives~/home~buHome])# ...
```

Changing the Retry Limit

A failed file transfer causes the shadow-copy rule to pause and then retry. The rule retries the directory (or directories) where the file(s) failed. By default, it pauses and retries five times before it stops and declares the shadow-copy run a failure. From gbl-ns-vol-shdwcp mode, you can use the retry attempts command to change the number of retries for the current rule:

```
retry attempts count
```

where ***count*** (optional, 0-2,147,483,647) is the number of retries that the rule should make if it is disconnected from the target volume.

The retries attempt to transfer all the directories that had transfer failures. Each retry invokes its own shadow-copy report.

For example, the following command sets 10 retries for the “insurSV” rule:

```
bstnA(gbl)# namespace insur volume /claims
bstnA(gbl-ns-vol[insur~/claims])# shadow-copy-rule insurSV
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# retry attempts 10
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# ...
```

Changing the Delay Between Retries

By default, a shadow-copy rule delays for 5 minutes (300 seconds) between its retries. You can use the retry delay command to change this delay:

```
retry delay seconds
```

where ***seconds*** (optional, 60-7200) is the time between retries.

For example, the following command sets a two-minute (or 120-second) delay for the “insurSV” rule:

```
bstnA(gbl)# namespace insur volume /claims
bstnA(gbl-ns-vol[insur~/claims])# shadow-copy-rule insurSV
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# retry delay 120
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# ...
```

Setting a Bandwidth Limit

You can use the bandwidth-limit command to limit the amount of network bandwidth used by the shadow-copy rule. Use this command from gbl-ns-vol-shdwcp mode:

bandwidth-limit *rate*[K|M|G|T]

where

rate (100,000-4,000,000,000,000) is the allowable bandwidth for shadow-copy transfers, and

K|M|G|T (optional) is the units for the rate: Kbps (1,000 bps), Mbps (1,000,000 bps), and so on. This default is bits-per-second (BPS).

The bandwidth limit applies to all transfers by the current shadow-copy rule.

For example, this command sequence limits transfers by the rule, “SVrule,” to five million BPS:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# bandwidth-limit 5M
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Removing any Bandwidth Limit

Use the no bandwidth-limit command to allow the rule to use unlimited bandwidth for its transfers:

no bandwidth-limit

For example, this command sequence removes any bandwidth limit from the “insurSV” rule:

```
bstnA(gbl)# namespace insur volume /claims
bstnA(gbl-ns-vol[insur~/claims])# shadow-copy-rule insurSV
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# no bandwidth-limit
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# ...
```

Supporting CIFS Options

This section describes the commands to support shadow copies from one CIFS-supporting volume to another. You can skip this section if the source volume is NFS-only.

Supporting Local Groups (CIFS)

A Windows filer can support *Global Groups*, which are managed by Domain Controllers, and/or *Local Groups*, which are unique to the filer. Local groups have their own Security IDs (SIDs), unknown to any other

Windows machine. When filers behind a source volume use local groups, they have SIDs that are unrecognized at the target volume's filers. This invalidates any Access Control Entries (ACEs) with these SIDs on the shadow volume; members of the local groups lose their privileges on the shadow volume. To resolve this problem, you must first prepare all of the target filers before you enable the shadow-copy rule:

- all local-group and local-user names must be configured on all filers behind *both* volumes, and
- all groups must contain the same users on those filers.

For example, if filers behind the source volume have a local group named "doctors," you must add a new "doctors" local group to all filers behind the target volume. The membership of these groups must match at all of the filers. This preparation is required so that all doctors can access the files at the source and shadow volumes.

This problem is similar for a volume with multiple CIFS filers behind it; recall *Supporting Filers with Local Groups*, on page 9-19.

◆ **Important**

If the source and shadow volumes are in two different Windows domains, SID translation may require customized preparation. Contact F5 Support before using this command for inter-domain shadow-copy rules.

Translating Local SIDs

After all local groups are duplicated on all source and destination filers, you must configure the shadow-copy rule to translate them. When SID translation is enabled, the rule finds a file's group name (such as "doctors") at the source volume, then looks up the SID for that group name at the destination filer. This introduces a slight performance penalty, but it ensures that doctors can access their files on both the source and shadow volumes. From gbl-ns-vol-shdwcp mode, use the sid-translation command to enable SID translations for the current shadow-copy rule:

```
sid-translation
```

For example, the following command sequence causes "SVrule" to translate all SIDs:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# sid-translation
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Failing On SID-Translation Errors

If a local group at a source filer is not configured at the target, SID translations fail for that local group. By default, the shadow-copy rule copies the original SID (a binary number) to the shadow volume's filer. The Access Control Entry (ACE) with the SID does not function at the shadow volume, though it is preserved; if the file is copied back to the source volume later, the SID will still be valid there.

To prevent the rule from copying a file that fails its SID translation, use the fail-on-errors option at the end of the command:

```
sid-translation fail-on-errors
```

For example:

```
bstnA(gbl)# namespace insur volume /claims
bstnA(gbl-ns-vol[insur~/claims])# shadow-copy-rule insurSV
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# sid-translation fail-on-errors
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# ...
```

◆ **Note**

Some filer servers can be configured to return an error for an invalid SID (STATUS_INVALID_SID, STATUS_INVALID_OWNER, and/or STATUS_INVALID_PRIMARY_GROUP) but accept the file or directory anyway. You may want to discount these errors from these particular file servers. You can set this up for each errant file server from gbl-ns-vol-shr mode, using the sid-translation ignore-sid-errors *command (recall* ***Ignoring SID Errors from the Filer (CIFS)****, on page 9-42).*

Disabling SID Translation

You can stop the shadow-copy rule from translating SIDs. This implies one of two scenarios:

- none of the filers behind the source volume use local groups, or
- clients do not require read access at the shadow volume.

This is the default for shadow-copy rules. Use no sid-translation to stop SID translation:

```
no sid-translation
```

For example, the following command sequence disables SID translation for the “insurSV” rule:

```
bstnA(gbl)# namespace insur volume /claims
bstnA(gbl-ns-vol[insur~/claims])# shadow-copy-rule insurSV
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# no sid-translation
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# ...
```

Copying Names That Fit the “8.3” Pattern (CIFS)

Early Windows file systems supported only short file and directory names, all following this “8.3” pattern:

filename[.*ext*]

where

filename is up to 8 characters,

. is optional, and

ext (optional) is up to 3 characters.

Newer file systems (such as NTFS) support names of any length, and also support a shorter *alternate name* for each file and directory. The alternate name is available for compatibility with older applications that only support the 8.3 naming scheme. For example, a file named "aReallyLongReport.doc" could have an alternate name of "~aRea001.doc." Client applications can use either name to access the same file, but directory listings only return the long name by default.

It is possible for a client to create a file or directory whose primary name matches some other file's alternate name. For example, a client could attempt to create a file named "~aRea001.doc" as its primary name. Each filer has its own set of alternate names, so it is possible for this name to co-exist with "aReallyLongReport.doc" on one filer but collide with it on another filer. To continue the example, suppose filer A uses "aReall~1.doc" as an alternate name for "aReallyLongReport.doc," and filer B assigns "~aRea001.doc" to the same file. A file with a primary name of "~aRea001.doc" has the potential to overwrite the "aReallyLongReport.doc" file on filer B.

If filer A is behind a source volume and filer B is behind its shadow volume, the shadow-copy rule cannot copy "~aRea001.doc" to the shadow volume without overwriting the "aReallyLongReport.doc" file. The shadow copy for this particular file cannot succeed without extra computation to avoid overwriting the original file. From gbl-ns-vol-shdwcp mode, you can use the cifs-8dot3-resolution command to make the current rule avoid issues with 8.3 names:

```
cifs-8dot3-resolution
```

For example, the following command sequence causes "insurSV" to take greater care with 8.3 names:

```
bstnA(gbl)# namespace insur
bstnA(gbl-ns[insur])# volume /claims
bstnA(gbl-ns-vol[insur~/claims])# shadow-copy-rule insurSV
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# cifs-8dot3-resolution
bstnA(gbl-ns-vol-shdwcp[insur~/claims~insurSV])# ...
```

This makes it possible for this shadow-copy rule, "insurSV," to successfully copy any file that matches the 8.3-naming pattern. We strongly recommend enabling this feature in any volume that supports CIFS.

Support for Multi-Protocol Volumes

The target volume must support all of the protocols in the source volume. That is, if the source volume supports only CIFS, the target volume must also be in a CIFS-only namespace. If the source volume supports CIFS and NFSv3, the target volume must support both of those protocols, too.

File-Attribute Translations

If the filers behind the source volume are from a different vendor than the filers behind the target volume, file attributes require some translation. (File attributes are the owner, group owner, permissions, timestamps, and other metadata associated with the file.) The issues and solutions for file-attribute replication are the same as those for file migrations in a multi-protocol volume. For a full discussion, refer to *Migrations in a Multi-Protocol Namespace*, on page 10-72 of the *ARX CLI Maintenance Guide*.

Managing the Target Volume's Directory Structure

Disabling Target Pruning (optional)

After the first shadow-copy run, the process *prunes* the directory tree on the target volume; that is, it removes all files and directories that were never on the source volume. For very large directory trees, the scanning for this can be very time-consuming. Some installations have a source volume and a shadow volume that they already know are identical; you can save time at those sites by disabling the prune-target feature. From gbl-ns-vol-shdwcp mode, use the no prune-target command to disable pruning:

```
no prune-target
```

This is only relevant before the first shadow copy. Pruning rarely occurs after the first run of the rule.

This only affects files that exist on the shadow volume and *never* existed on the source volume. These are files or directories that were imported into the shadow volume and had no counterparts in the source volume. If a file or directory exists on both and a client deletes it from the source volume, the shadow-copy rule deletes it from the shadow volume, too: the prune-target command has no effect on these standard deletions.

For example, the following command sequence disables pruning for the shadow-copy rule, SVrule:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# no prune-target
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Enabling Pruning

Target-pruning is enabled by default, to ensure that the source and shadow volumes match after the first shadow copy. From gbl-ns-vol-shdwcp mode, use the prune-target command to re instate this default:

```
prune-target
```

For example, the following command sequence enables pruning for the 'bkup' rule.

```
prtlndA(gbl)# namespace testns volume /users
prtlndA(gbl-ns-vol[testns~/users])# shadow-copy-rule bkup
prtlndA(gbl-ns-vol-shdwcp[testns~/users~bkup])# prune-target
prtlndA(gbl-ns-vol-shdwcp[testns~/users~bkup])# ...
```

Publishing All Files as a Group (optional)

By default, the shadow-copy rule publishes all of its successfully-transferred files even if some of the file transfers fail. Some applications (such as a CAD application) require all files to be synchronized; a single missing file can cause these applications to fail. You can configure a shadow-copy rule so that no files are published in the shadow volume if any file fails to transfer properly.

To publish (or not publish) all files as a group, use the publish group command in gbl-ns-vol-shdwcp mode:

```
publish group
```

For example:

```
bstnA(gbl)# namespace ns3 volume /cad1
bstnA(gbl-ns-vol[ns3~/cad1])# shadow-copy-rule cpRemote
bstnA(gbl-ns-vol-shdwcp[ns3~/cad1~cpRemote])# publish group
bstnA(gbl-ns-vol-shdwcp[ns3~/cad1~cpRemote])# ...
```

Publishing Individual Files

By default, a shadow-copy rule publishes all files that successfully transfer, whether or not some of the transfers fail. To return to this default, use the publish individual command:

```
publish individual
```

For example:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# publish individual
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Changing the Threshold for Delta Transfers (optional)

The shadow-copy rule generally copies the *delta* for each file that changed. That is, the source and shadow files are compared on a block-by-block basis, and only the blocks that changed are sent to the shadow volume. For large files, this saves network bandwidth. As files get smaller, the bandwidth savings can be offset by the cost of computation time.

By default, the shadow-copy rule performs delta transfers for files 100 MegaBytes or larger. Files smaller than 100 MegaBytes are transferred in their entirety. Note that both versions of the file (source and shadow) must exceed the delta threshold for the delta transfer to take place.

From gbl-ns-vol-shdwcp mode, use the delta-threshold command to set a different threshold:

```
delta-threshold minimum-size[k|M|G|T]
```

where

minimum-size (1-18,446,744,073,709,551,615) is the minimum size of a file that is eligible for delta transfer, and

k|M|G|T (optional) sets the size units: **k**ilobytes, **M**egabytes, **G**igabytes, or **T**erabytes. The default is bytes if you omit this. All of these values are 2-based, so a kilobyte is 1024 bytes, a megabytes is 1024*1024, and so on. There can be no spaces between the ***minimum-size*** and the unit letter; **20M** is correct, but **20 M** is invalid.

For example, the following command sequence sets a threshold of 500 megabytes for delta transfers:

```
bstnA(gbl)# namespace archives volume /home
bstnA(gbl-ns-vol[archives~/home])# shadow-copy-rule buHome
bstnA(gbl-ns-vol-shdwcp[archives~/home~buHome])# delta-threshold 500M
bstnA(gbl-ns-vol-shdwcp[archives~/home~buHome])# ...
```

Reverting to the Default

To reset the threshold back to the default, 100 megabytes, use no delta-threshold:

```
no delta-threshold
```

For example:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# no delta-threshold
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Enabling the Shadow-Copy Rule

The final step in configuring the shadow-copy rule is to enable it. By default, the rule is disabled and ignored by policy software. Use the enable command to enable the rule:

```
enable
```

For example, the following command sequence enables the rule, “SVrule:”

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule SVrule
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# enable
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~SVrule])# ...
```

Disabling the Rule

Disabling the rule stops all file replication. Use no enable from gbl-ns-vol-shdwcp mode to disable a shadow-copy rule:

```
no enable
```

For example, the following command sequence disables the “buHome” rule:

```
bstnA(gbl)# namespace archives volume /home
bstnA(gbl-ns-vol[archives~/home])# shadow-copy-rule buHome
bstnA(gbl-ns-vol-shdwcp[archives~/home~buHome])# no enable
bstnA(gbl-ns-vol-shdwcp[archives~/home~buHome])# ...
```

Showing Shadow-Copy Status

Use the show shadow command to view the high-level status of the latest (or current) shadow-copy run:

show shadow

For every shadow-copy rule in every namespace, this shows the current progress of all shadow copies. Each rule has an overview section, a Target Status section listing all of the shadow volume targets, and two or more sections with more detail. The additional sections describe the most time-consuming parts of a shadow copy:

1. Copy Phase, where files are copied from the source volume to a staging area in the shadow volume.
2. Publishing Phase, where the files move from the staging area into their proper places in the directory tree.

For example, this shows two shadow-copy rules:

```
bstnA(gbl)# show shadow

Shadow Copy Status
==================

    Namespace                                     :   wwmed
    Source Volume                                 :   /acct
    Shadow Rule                                   :   SVrule
    Report File                                   :   SVetc_201009160410.rpt
    Fileset                                       :   worthSaving (Files and Directories)
    Delta Threshold                               :   100 MB
    Publishing Mode                               :   Individual
    CIFS SID Translation                          :   Ignore Errors
    Shared Access Allowed                         :   Yes
    Inline Notifications                          :   Yes
    Bandwidth Limit                               :   5.0 Mb/s (625.0 kB/s)
    Retry Attempts                                :   5
    Retry Delay                                   :   300 seconds
    Database Location                             :   metadata-share

    ============================================================================
      Processing Started                          :   Sep 16 04:00
      Processing Completed                        :   Sep 16 04:10
      Elapsed Time                                :   00:10:46
      Operating Mode                              :   Inline notification
      Current Phase                               :   Completed

      Target Information
      ------------------
        prtlndA: nemed:/acctShdw/                 :   Error(s) detected

      Copy Phase Information
      ----------------------
        Phase Started                             :   Sep 16 04:10
        Phase Completed                           :   Sep 16 04:10
        Elapsed Time                              :   00:00:00
        Total Files/Directories                   :   4,848
        Files/Directories Scanned                 :   0
        Files/Directories Skipped                 :   0
        Files/Directories Processed               :   0
            Identical Files/Directories           :   0
            New Files/Directories                 :   0
            Updated Files/Directories             :   0
        Full Update Bytes Sent                    :   0 (0 B)
        Delta Update Bytes Sent                   :   0 (0 B)
            Effective Data Bytes                  :   0 (0 B)


Shadow Copy Status
==================

    Namespace                                     :   insur
    Source Volume                                 :   /claims
    Shadow Rule                                   :   insurSV
    Report File                                   :   insurSV_201009160410.rpt
    Fileset                                       :   worthSaving (Files and Directories)
    Delta Threshold                               :   100 MB
    Publishing Mode                               :   Individual
    CIFS SID Translation                          :   Disabled
    Shared Access Allowed                         :   Yes
    Inline Notifications                          :   Yes
    Prune Target                                  :   Yes
    Retry Attempts                                :   10
    Retry Delay                                   :   120 seconds
    Database Location                             :   metadata-share
    CIFS 8dot3 Resolution                         :   Yes

    ============================================================================
      Processing Started                          :   Sep 16 04:00
      Processing Completed                        :   Sep 16 04:10
      Elapsed Time                                :   00:10:51
      Total Elapsed Time                          :   00:18:56
      Operating Mode                              :   Full tree walk
      Tree Walk Reason                            :   Initial run (resumed)
      Current Phase                               :   Completed

      Target Information
      ------------------
        prtlndA: insur_bkup:/insurShdw/           :   Error(s) detected

      Copy Phase Information
      ----------------------
        Phase Started                             :   Sep 16 04:10
        Phase Completed                           :   Sep 16 04:10
        Elapsed Time                              :   00:00:07
        Average Transmission Rate                 :   7.3 kb/s (915 B/s)
        Total Files/Directories                   :   196
        Files/Directories Scanned                 :   175
        Files/Directories Skipped                 :   21
            Other Failure Conditions              :   21
        Files/Directories Processed               :   189
            Identical Files/Directories           :   21
            New Files/Directories                 :   168
            Updated Files/Directories             :   0
        Full Update Bytes Sent                    :   187,401,844 (178 MB)
        Delta Update Bytes Sent                   :   0 (0 B)
            Effective Data Bytes                  :   0 (0 B)

      Publishing Phase Information
      ----------------------------
        Database Records Scanned                  :   0
        Files/Directories Published              :   183
            New Files/Directories                :   167
            Renamed Files/Directories            :   0
            Updated Files/Directories            :   16
            Removed Files/Directories            :   0
...

bstnA(gbl)# ...
```

Focusing on One Namespace

To specify one namespace, specify its name at the end of the command:

```
show shadow namespace
```

where ***namespace*** (1-30 characters) identifies a namespace with one or more source volumes.

For example:

```
bstnA(gbl)# show shadow wwmed
...
```

Focusing on One Volume

After the optional namespace name, you can add a volume path to specify one source volume:

```
show shadow namespace vol-path
```

where

namespace (1-30 characters) identifies the source namespace, and

vol-path (1-1024 characters) identifies the source volume.

For example:

```
bstnA(gbl)# show shadow wwmed /acct
...
```

Focusing on One Rule

Add the rule name to specify one shadow-copy rule:

```
show shadow namespace vol-path rule-name
```

where

namespace (1-30 characters) identifies the source namespace,

vol-path (1-1024 characters) identifies the source volume, and

rule-name (1-64 characters) is the name of the rule.

For example:

```
bstnA(gbl)# show shadow wwmed /acct SVrule
...
```

Monitoring the Progress of a Shadow Copy

Use show reports type SCp to find the report for the shadow-copy rule, then use tail reports *report-name* follow to monitor the report as it is created.

◆ **Note**

All shadow-copy reports are written to the ARX that hosts the source volume.

For example, the following command sequence finds a report on the “bstnA” chassis and tails it:

```
bstnA(gbl)# show reports type SCp
bstnA# show reports type SCp


  reports
    Codes: SCp=Shadow Copy
      SVetc_200909140134.rpt  Sep 14 01:41  594 kB        SCp  RUNNING: 809 in 00:04:36
      insurSV_200909140150.rpt Sep 14 01:50  14 kB        SCp  DONE: 191 in 00:02:07

bstnA(gbl)# tail reports SVetc_200909140134.rpt follow
    /shellCmds_winXP.fm
prtlndA: nemed:/acctShdw/: Full update (275,584 bytes sent)
    /shellCmds_winXP.fm                                                                  275,456
New(1)
    /troubleShootNet.fm.lck
prtlndA: nemed:/acctShdw/: Full update (212 bytes sent)
    /troubleShootNet.fm.lck                                                                   84
New(1)
    /navNest.js
prtlndA: nemed:/acctShdw/: Full update (18,484 bytes sent)
    /navNest.js                                                                           18,353
New(1)
    /wwmed.css
prtlndA: nemed:/acctShdw/: Full update (1,024 bytes sent)
    /wwmed.css                                                                               895
New(1)
    /ci.2.1.4
prtlndA: nemed:/acctShdw/: Full update (88 bytes sent)
    /ci.2.1.4                                                                                 47
New(1)
...
```

You can also use show reports *report-name* to read a report, or grep to find a string in the report.

Report Format

The shadow-copy report shows the progress of tree walks and replications (the “Copy Phase” in the show shadow output) on a file-by-file basis. The standard output from show shadow (discussed above) appears at the beginning and end of this report.

For example,

```
bstnA(gbl)# show reports SVetc_200909140134.rpt
bstnA# show reports SVetc_201204110400.rpt
```

```
**** Shadow Copy Report: Started at 04/11/2012 04:00:00 -0400 ****
**** Software Version: 6.02.000.14353 (Apr  6 2012 20:12:43) [nbuilds]
**** Hardware Platform: ARX-4000
**** Report Destination:

Shadow Copy Status
==================

    Namespace                                      :   wwmed
    Source Volume                                  :   /acct
    Shadow Rule                                    :   SVrule
    Fileset                                        :   worthSaving (Files and Directories)
    Delta Threshold                                :   100 MB
    Publishing Mode                                :   Individual
    CIFS SID Translation                           :   Ignore Errors
    Shared Access Allowed                          :   Yes
    Inline Notifications                           :   Yes
    Bandwidth Limit                                :   5.0 Mb/s (625.0 kB/s)
    Retry Attempts                                 :   5
    Retry Delay                                    :   300 seconds
    Database Location                              :   metadata-share

    =============================================================================
      Processing Started                           :   04/11 04:00
      Processing Completed                         :   04/11 04:00
      Elapsed Time                                 :   00:00:04
      Operating Mode                               :   Inline notification
      Current Phase                                :   Completed

      Target Information
      ------------------
        prtlndA: nemed:/acctShdw/                  :   Successful

      Copy Phase Information
      ----------------------
        Phase Started                              :   04/11 04:00
        Phase Completed                            :   04/11 04:00
        Elapsed Time                               :   00:00:00
        Total Files/Directories                    :   4,848
        Files/Directories Scanned                  :   0
        Files/Directories Skipped                  :   0
        Files/Directories Processed                :   0
            Identical Files/Directories            :   0
            New Files/Directories                  :   0
            Updated Files/Directories              :   0
        Full Update Bytes Sent                     :   0 (0 B)
        Delta Update Bytes Sent                    :   0 (0 B)
            Effective Data Bytes                   :   0 (0 B)

      Publishing Phase Information
      ----------------------------
        Phase Started                              :   04/11 04:00
        Phase Completed                            :   04/11 04:00
        Elapsed Time                               :   00:00:00
        Database Records Scanned                   :   110
        Files/Directories Published                :   0
            New Files/Directories                  :   0
            Renamed Files/Directories              :   0
            Updated Files/Directories              :   0
            Removed Files/Directories              :   0

Total processed:               0
```

```
Elapsed time:           00:00:04
**** Shadow Copy Report: DONE at 04/11/2012 04:00:04 -0400 ****
```

Reformatting the Report

You can use the copy reports command to duplicate the report in a different format, XML or CSV (Comma-Separated Values). The duplicate's file extension indicates the desired format: to convert to XML, use a .xml extension; to convert to CSV, use .csv. The copy command is available in priv-exec mode:

copy reports *source destination*[.xml|.csv]

where

source (1-255 characters) specifies report to copy,

destination (1-255 characters) is the name of the duplicate report, and

[.xml|.csv] chooses the duplicate's format (XML or CSV, respectively).

A .txt or .rpt extension keeps the destination report in plain-text format.

For example, the following command sequence exits to priv-exec mode and creates an XML copy of a shadow report:

```
bstnA(gbl)# exit
bstnA# copy reports SVetc_200909140134.rpt SVetc_9-14.xml
bstnA# ...
```

Truncating the Report

To conserve CPU cycles and/or internal-disk space, you may want to stop a shadow-copy report before it is finished. An oversized, CPU-intensive report could possibly have an effect on namespace performance. From priv-exec mode, use the truncate-report command to stop all report processing and truncate the report file:

truncate-report *name*

where ***name*** (1-255 characters) specifies report to truncate.

This only stops the policy engine from writing to the report; the shadow-copy processing continues.

For example, the following command finds a running shadow-copy report and truncates it:

```
bstnA(gbl)# show reports type SCp

  reports
    Codes: SCp=Shadow Copy
      SVetc_200802270143.rpt  Feb 27 01:52  920k        SCp  RUNNING: 1356 in 00:06:53
      SVetc_200802270200.rpt  Feb 27 02:00  6.0k        SCp  DONE: 0 in 00:00:05
      insurSV_200802270143.rpt Feb 27 01:44   76k        SCp  DONE: 129 in 00:01:16
      insurSV_200802270200.rpt Feb 27 02:00  7.4k        SCp  DONE: 0 in 00:00:13

bstnA(gbl)# end
bstnA# truncate-report SVetc_200802270143.rpt
bstnA# ...
```

Copying the Source Volume to Multiple Targets

To copy a source volume to more than one target, create a separate shadow-copy rule for each target. Each shadow-copy rule can operate on its own schedule, or they can all use the same schedule. For example, the following command sequence sets up a second shadow-copy rule for the “wwmed~/acct” volume, which goes to a different target switch using the same schedule:

```
bstnA(gbl)# namespace wwmed volume /acct
bstnA(gbl-ns-vol[wwmed~/acct])# shadow-copy-rule bkup
This will create a new policy object.

Create object 'bkup'? [yes/no] yes
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~bkup])# from fileset worthSaving
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~bkup])# target hostname provA namespace wwmedBk volume /acBk
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~bkup])# schedule hourly
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~bkup])# report bkup
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~bkup])# enable
bstnA(gbl-ns-vol-shdwcp[wwmed~/acct~bkup])# ...
```

Removing the Shadow-Copy Rule

You can remove a shadow-copy rule to both disable it and delete its configuration. Use the no form of the shadow-copy-rule command to remove a shadow-copy rule:

```
no shadow-copy-rule name
```

where ***name*** (1-1024 characters) identifies the rule to be removed.

For example, the following command sequence removes the shadow-copy rule, “buHome,” from the “/home” volume:

```
bstnA(gbl)# namespace archives volume /home
bstnA(gbl-ns-vol[archives~/home])# no shadow-copy-rule buHome
bstnA(gbl-ns-vol[archives~/home])# ...
```

Index