Tivoli Storage Manager
for AIX

Administrator's Guide

Version 5 Release 1

GC32-0768-00

Tivoli Storage Manager
for AIX

Administrator's Guide

Version 5 Release 1

GC32-0768-00

Note!

Before using this information and the product it supports, be sure to read the general information under "Appendix C. Notices" on page 619.

First Edition (March 2002)

This edition applies to Version 5 Release 1 of the Tivoli Storage Manager for AIX® (product numbers 5698-ISM, 5698-ISE) and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

Order publications through your sales representative or the branch office serving your locality.

Your feedback is important in helping to provide the most accurate and high-quality information. If you have comments about this book or any other Tivoli Storage Manager documentation, please see "Contacting Customer Support" on page xv.

© Copyright International Business Machines Corporation 1993, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 1993, 2002

Chapter 5. Configuring Storage Devices 67

Chapter 6. Setting Up Tivoli Data Protection for NDMP 97

Chapter 7. Managing Removable Media Operations. 113

Part 3. Managing Client Operations 225

Chapter 10. Adding Client Nodes . . . 227

Chapter 11. Managing Client Nodes 235

Chapter 12. Implementing Policies for Client Data 269

Part 4. Maintaining the Server . . . 347

Chapter 16. Managing Server Operations . . . 349

Chapter 17. Automating Server Operations . . . 367

Chapter 18. Managing the Database and Recovery Log . . . 383

Chapter 19. Monitoring the Tivoli Storage Manager Server . . . 401

Chapter 20. Working with a Network of Tivoli Storage Manager Servers . . 431

Chapter 21. Exporting and Importing Data 477

Part 5. Protecting the Server . . . 497

Chapter 22. Protecting and Recovering Your Server 499

Preface

Tivoli® Storage Manager is a client/server program that provides storage management solutions to customers in a multivendor computer environment. Tivoli Storage Manager provides an automated, centrally scheduled, policy-managed backup, archive, and space-management facility for file servers and workstations.

Who Should Read This Publication

This reference is intended for anyone who is registered as an administrator. A single administrator can manage Tivoli Storage Manager; however, several people can share administrative responsibilities.

You can invoke all of the administrator commands that you need to operate and maintain Tivoli Storage Manager from:

- A workstation connected to the server
- A workstation with a Web browser that meets the requirements specified in the *Tivoli Storage Manager Quick Start*

What You Should Know before Reading This Publication

You should be familiar with the operating system on which the server resides and the communication protocols required for the client/server environment.

For information on installing Tivoli Storage Manager, see the *Tivoli Storage Manager Quick Start*.

You also need to understand the storage management practices of your organization, such as how you are currently backing up your workstation files and how you are using storage devices.

Tivoli Storage Manager Web Site

Technical support information and publications are available at the following address:

http://www.tivoli.com/support/storage_mgr/tivolimain.html

Conventions Used in This Book

To help you recognize where example commands are to be entered, this book uses the following conventions:

- Command to be entered on the AIX command line:

```
> dsmadmc
```

- Command to be entered on the command line of an administrative client:

```
query devclass
```

© Copyright IBM Corp. 1993, 2002

Tivoli Storage Manager Publications

The following table lists Tivoli Storage Manager server publications.

Publication Title	Order Number
Tivoli Storage Management Products License Information	GH09-4572
Tivoli Storage Manager Messages	GC32-0767
Tivoli Storage Manager for AIX Administrator's Guide	GC32-0768
Tivoli Storage Manager for AIX Administrator's Reference	GC32-0769
Tivoli Storage Manager for AIX Quick Start	GC32-0770

The following table lists Tivoli Storage Manager storage agent publications.

Publication Title	Order Number
Tivoli Storage Manager for AIX Managed System for SAN Storage Agent User's Guide	GC32-0771
Tivoli Storage Manager for HP-UX Managed System for SAN Storage Agent User's Guide	GC32-0727
Tivoli Storage Manager for Sun Solaris Managed System for SAN Storage Agent User's Guide	GC32-0781
Tivoli Storage Manager for Windows Managed System for SAN Storage Agent User's Guide	GC32-0785

The following table lists the Tivoli Storage Manager client publications.

Publication Title	Order Number
Tivoli Space Manager for UNIX: Using the Hierarchical Storage Management Clients	GC32-0794
Tivoli Storage Manager for Macintosh: Backup-Archive Installation and User's Guide	GC32-0787
Tivoli Storage Manager for NetWare: Backup-Archive Installation and User's Guide	GC32-0786
Tivoli Storage Manager for UNIX: Backup-Archive Installation and User's Guide	GC32-0789
Tivoli Storage Manager for Windows: Backup-Archive Installation and User's Guide	GC32-0788
Tivoli Storage Manager Trace Facility Guide	GC32-0792
Tivoli Storage Manager Using the Application Program Interface	GC32-0793

The following table lists Tivoli Data Protection publications.

Publication Title	Order Number
Tivoli Data Protection for EMC Symmetrix for R/3 Installation and User's Guide	SC33-6386
Tivoli Data Protection for IBM ESS for Oracle Databases DB2 UDB Installation and User's Guide	SH26-4125
Tivoli Data Protection for IBM ESS for R/3 Installation and User's Guide	SC33-6387
Tivoli Data Protection for Informix Installation and User's Guide	SH26-4095

Publication Title	Order Number
Tivoli Data Protection for Lotus Domino for UNIX Installation and User's Guide	SH26-4088
Tivoli Data Protection for Lotus Domino for Windows Installation	GC26-7320
Tivoli Data Protection for Lotus Domino, S/390 Edition Licensed Program Specifications	GC26-7305
Tivoli Data Protection for Lotus Notes on AIX Installation and User's Guide	SH26-4067
Tivoli Data Protection for Lotus Notes on Windows NT Installation and User's Guide	SH26-4065
Tivoli Data Protection for Microsoft Exchange Server Installation and User's Guide	SH26-4110
Tivoli Data Protection for Microsoft SQL Server Installation and User's Guide	SH26-4111
Tivoli Data Protection for Oracle for UNIX Installation and User's Guide	SH26-4112
Tivoli Data Protection for Oracle for Windows Installation and User's Guide	SH26-4113
Tivoli Data Protection for R/3 Installation and User's Guide for Oracle	SC33-6340
Tivoli Data Protection for R/3 Installation and User's Guide for DB2 UDB	SC33-6341
Tivoli Data Protection for R/3 Installation and User's Guide	SC33-6389
Tivoli Data Protection for WebSphere Application Server Installation and User's Guide	SC33-6399
Tivoli Data Protection for Workgroups for NetWare User's Guide	GC32-0444

Related IBM® Hardware Products Publications

The following table lists related IBM hardware products publications.

Title	Order Number
IBM Magstar 3494 Tape Library Introduction and Planning Guide	GA32-0279
IBM 3490E Model E01 and E11 User's Guide	GA32-0298
IBM Magstar MP 3570 Tape Subsystem Operator's Guide	GA32-0345
IBM SCSI Tape Drive, Medium Changer, and Library Device Drivers: Installation and User's Guide	GC35-0154
IBM TotalStorage Enterprise Tape System 3590 Operator Guide	GA32-0330
IBM Magstar 3494 Tape Library Dataserver Operator Guide	GA32-0280

IBM International Technical Support Center Publications (Redbooks™)

The International Technical Support Center (ITSC) publishes Redbooks, which are books on specialized topics such as using Tivoli Storage Manager to back up databases. You can order publications through your IBM representative or the IBM branch office serving your locality. You can also search for and order books of interest to you at the IBM Redbooks Web site at this address:

http://www.redbooks.ibm.com/redbooks

Contacting Customer Support

For support for this or any Tivoli product, you can contact Tivoli Customer Support in one of the following ways:

- Visit the Tivoli Storage Manager technical support Web site at http://www.tivoli.com/support/storage_mgr/tivolimain.html.
- Submit a problem management record (PMR) electronically at **IBMSERV/IBMLINK**. You can access IBMLINK at http://www2.ibmlink.ibm.com.
- Submit a problem management record (PMR) electronically at http://www.tivoli.com/support. See "Reporting a Problem" for details.
- Send e-mail to support@tivoli.com.

Customers in the United States can also call 1-800-TIVOLI8 (1-800-848-6548). For product numbers 5697-TS9, 5697-DRS or 5697-DPM call 1-800-237-5511.

International customers should consult the Web site for customer support telephone numbers.

You can also review the *Customer Support Handbook*, which is available on our Web site at http://www.tivoli.com/support/handbook/.

When you contact Tivoli Customer Support, be prepared to provide identification information for your company so that support personnel can readily assist you. Company identification information may also be needed to access various online services available on the Web site.

The support Web site offers extensive information, including a guide to support services (the Customer Support Handbook); frequently asked questions (FAQs); and documentation for all Tivoli products, including Release Notes, Redbooks, and Whitepapers. The documentation for some product releases is available in both PDF and HTML formats. Translated documents are also available for some product releases.

You can order documentation by e-mail at swdist@tivoli.com. Please provide the publication number, part number, or order number of the desired document. Alternatively, you can provide the document title, version number, and date of publication.

We are very interested in hearing about your experience with Tivoli products and documentation. We also welcome your suggestions for improvements. If you have comments or suggestions about our documentation, please contact us in one of the following ways:

- Send e-mail to pubs@tivoli.com.
- Complete our customer feedback survey at http://www.tivoli.com/support/feedback/.

Reporting a Problem

Please have the following information ready when you report a problem:

- The Tivoli Storage Manager server version, release, modification, and service level number. You can get this information by entering the `QUERY STATUS` command at the TSM command line.
- The Tivoli Storage Manager client version, release, modification, and service level number. You can get this information by entering `dsmc` at the command line.

- The communication protocol (for example, TCP/IP), version, and release number you are using.
- The activity you were doing when the problem occurred, listing the steps you followed before the problem occurred.
- The exact text of any error messages.

Translations

Selected Tivoli Storage Manager publications have been translated into languages other than American English. Contact your sales representative for more information about the translated publications and whether these translations are available in your country.

Summary of Changes for Tivoli Storage Manager Version 5

This section summarizes changes that have been made to the Tivoli Storage Manager product and this publication.

Technical Changes for Version 5 Release 1—March 2002

The following changes have been made to the product for this edition:

Changes in Defining Drives and Libraries

Device special file names and external library managers are now specified in the DEFINE PATH and UPDATE PATH commands, rather than in the DEFINE DRIVE, UPDATE DRIVE, DEFINE LIBRARY, and UPDATE LIBRARY commands.

See "Chapter 5. Configuring Storage Devices" on page 67. Also see *Tivoli Storage Manager Administrator's Reference*.

Moving Data by Node

You can use the MOVE NODEDATA command to move data in a sequential-access storage pool for one or more nodes, or move selected file spaces for a single node. You can also use MOVE NODEDATA to move data to another storage pool.

See "Chapter 9. Managing Storage Pools and Volumes" on page 155.

Support for Simultaneous Writes to Primary and Copy Storage Pools

You can specify copy storage pools in a primary storage pool definition. When a client backs up, archives, or migrates a file, the file is written to the primary storage pool and is simultaneously stored into each copy storage pool.

See "Chapter 9. Managing Storage Pools and Volumes" on page 155.

High Availability Cluster Multiprocessing

Tivoli Storage Manager can now use High Availability Cluster Multiprocessing (HACMP). HACMP provides the leading AIX-based clustering solution, which allows automatic system recovery during system failure detection. By using HACMP together with Tivoli Storage Manager, you can ensure server availability.

Tivoli Data Protection for New Network Data Management Protocol Support

New Network Data Management Protocol (NDMP) support now extends to the AIX (32-bit and 64-bit) Tivoli Storage Manager server platform. The new Tivoli Data Protection for NDMP product supports NDMP backup and restore for network-attached storage (NAS) file servers from Network Appliance. NDMP allows a network storage-management application to control the backup and restore of an NDMP-compliant file server without installing third-party software on that server. The NAS file server does not require installation of TSM software. The TSM server uses NDMP to connect to the NAS file server to initiate, control, and monitor a file system backup or restore operation. The NDMP support for NAS file servers enables higher performance backup to tape devices without moving the data over the LAN. TDP for NDMP is a separately priced and licensed product.

© Copyright IBM Corp. 1993, 2002

See “Chapter 6. Setting Up Tivoli Data Protection for NDMP” on page 97.

Data Validation with Cyclic Redundancy Checking

Tivoli Storage Manager provides the option of specifying whether a cyclic redundancy check (CRC) is performed during a client session with the server, or for storage pools. The server validates the data by using a cyclic redundancy check which can help identify data corruption. Data validation can be enabled for one or all of the following:

- Tivoli Storage Manager client nodes at Version 5.1. See “Validating a Node’s Data” on page 315.
- Tivoli Storage Manager storage agents at Version 5.1. See *Tivoli Storage Manager Managed System for SAN Storage Agent User’s Guide* for more information.
- Storage pools. See “Data Validation During Audit Volume Processing” on page 531 and “Chapter 9. Managing Storage Pools and Volumes” on page 155.

New Licensing Method

The new licensing method enables you to register the exact number of licenses that are required, rather than in increments of 1, 5, 10, and 50.

See “Registering Licensed Features” on page 350.

Server Performance Enhancements

There are two new Tivoli Storage Manager performance enhancements:

- AIX Asynchronous I/O Support. This feature is available via a new option in the server options file.
- AIX Direct I/O Support. This feature is available via a new option in the server options file.

See “Using Server Performance Options” on page 363.

Part 1. Tivoli Storage Manager Basics

© Copyright IBM Corp. 1993, 2002

Chapter 1. Introducing Tivoli Storage Manager

Tivoli Storage Manager is an enterprise-wide storage management application. It provides automated storage management services to workstations, personal computers, and file servers from a variety of vendors, with a variety of operating systems. Tivoli Storage Manager includes the following components:

Server

Server program
The server program provides backup, archive, and space management services to the clients.

You can set up multiple servers in your enterprise network to balance storage, processor, and network resources.

Administrative interface
The administrative interface allows administrators to control and monitor server activities, define management policies for clients, and set up schedules to provide services to clients at regular intervals. Administrative interfaces available include a command-line administrative client and a Web browser interface. Tivoli Storage Manager allows you to manage and control multiple servers from a single interface that runs in a Web browser.

Server database and recovery log
The Tivoli Storage Manager server uses a database to track information about server storage, clients, client data, policy, and schedules. The server uses the recovery log as a scratch pad for the database, recording information about client and server actions while the actions are being performed.

Server storage
The server can write data to hard disk drives, disk arrays and subsystems, stand-alone tape drives, tape libraries, and other forms of random- and sequential-access storage. The media that the server uses are grouped into *storage pools*. The storage devices can be connected directly to the server, or connected via local area network (LAN) or storage area network (SAN).

Client Nodes
A client node can be a workstation, a personal computer, a file server, a network-attached storage (NAS) file server, or even another Tivoli Storage Manager server. The client node has Tivoli Storage Manager client software installed (except for NAS file servers using NDMP). A client node is registered with the server.

Backup-archive client
The backup-archive client allows users to maintain backup versions of files, which they can restore if the original files are lost or damaged. Users can also archive files for long-term storage and retrieve the archived files when necessary. Users themselves or administrators can register workstations and file servers as client nodes with a Tivoli Storage Manager server.

The storage agent is an optional component that may also be installed on a system that is a client node. The storage agent enables LAN-free data movement for client operations.

© Copyright IBM Corp. 1993, 2002

Application program interface (API)
The API allows you to enhance existing applications to use the backup, archive, restore, and retrieve services that Tivoli Storage Manager provides. Tivoli Storage Manager API clients can register as client nodes with a Tivoli Storage Manager server.

Tivoli Storage Manager supports separately licensed, optional products. These include:

Tivoli Data Protection for applications (application clients)
Application clients allow users to perform online backups of data for applications such as database programs. After the application program initiates a backup or restore, the application client acts as the interface to Tivoli Storage Manager. The Tivoli Storage Manager server then applies its storage management functions to the data. The application client can perform its functions while application users are working, with minimal disruption.

Tivoli Data Protection for NDMP
The Tivoli Data Protection for NDMP product allows you to use Tivoli Storage Manager with the Network Data Management Protocol (NDMP) to back up and restore file systems stored on a network-attached storage (NAS) file server. The data on the NAS file server is backed up to a tape library. See "Chapter 6. Setting Up Tivoli Data Protection for NDMP" on page 97 for more information.

Tivoli Space Manager
Tivoli Space Manager provides space management services for workstations on some platforms. Tivoli Space Manager automatically migrates files that are less frequently used to server storage, freeing space on the workstation. The migrated files are also called *space-managed files*. Users can recall space-managed files automatically simply by accessing them as they normally would from the workstation. Tivoli Space Manager is also known as the hierarchical storage management (HSM) client.

For information about supported operating systems for clients, see the Tivoli Storage Manager Web site:

http://www.tivoli.com/support/storage_mgr/tivolimain.html

Client programs such as the backup-archive client and the HSM client (space manager) are installed on systems that are connected through a LAN and are registered as client nodes. From these client nodes, users can back up, archive, or migrate files to the server.

The following sections present key concepts and information about Tivoli Storage Manager. The sections describe how Tivoli Storage Manager manages client files based on information provided in administrator-defined policies, and manages devices and media based on information provided in administrator-defined Tivoli Storage Manager storage objects.

The final section gives an overview of tasks for the administrator of the server, including options for configuring the server and how to maintain the server.

Concepts:
"How Tivoli Storage Manager Stores Client Data"
"How the Server Manages Storage" on page 14
"Configuring and Maintaining the Server" on page 16

How Tivoli Storage Manager Stores Client Data

Tivoli Storage Manager policies are rules that determine how the client data is stored and managed. The rules include where the data is initially stored, how many backup versions are kept, how long archive copies are kept, and so on. You can have multiple policies and assign the different policies as needed to specific clients, or even to specific files.

Policy assigns a location in server storage where data is initially stored. Server storage is divided into storage pools that are groups of storage volumes. Server storage can include hard disk, optical, and tape volumes.

When you install Tivoli Storage Manager, you have a default policy that you can use. For details about this default policy, see "The Standard Policy" on page 271. You can modify this policy and define other policies.

Clients use Tivoli Storage Manager to store data for any of the following purposes:

Backup and restore

The backup process copies data from client workstations to server storage to ensure against loss of data that is regularly changed. The server retains versions of a file according to policy, and replaces older versions of the file with newer versions. Policy includes the number of versions and the retention time for versions.

A client can restore the most recent version of a file, or can restore earlier versions.

Archive and retrieve

The archive process copies data from client workstations to server storage for long-term storage. The process can optionally delete the archived files from the client workstations. The server retains archive copies according to the policy for archive retention time. A client can retrieve an archived copy of a file.

Instant archive and rapid recovery

Instant archive is the creation of a complete set of backed-up files for a client. The set of files is called a *backup set*. A backup set is created on the server from the most recently backed-up files that are already stored in server storage for the client. Policy for the backup set consists of the retention time that you choose when you create the backup set.

You can copy a backup set onto compatible portable media, which can then be taken directly to the client for rapid recovery without the use of a network and without having to communicate with the Tivoli Storage Manager server.

Migration and recall

Migration, a function of the Tivoli Space Manager program, frees up client storage space by copying files from workstations to server storage. On the client, the Tivoli Space Manager program replaces the original file with a stub file that points to the original in server storage. Files are recalled to the workstations when needed.

This process is also called hierarchical storage management (HSM). Once configured, the process is transparent to the users. Files are migrated and recalled automatically.

Policy determines when files are considered for automatic migration. On the UNIX systems that support the Tivoli Space Manager program, policies determine whether files must be backed up to the server before being migrated. Space management is also integrated with backup. If the file to be backed up is already migrated to server storage, the file is backed up from there.

Figure 1 shows how policy is part of the Tivoli Storage Manager process for storing client data.

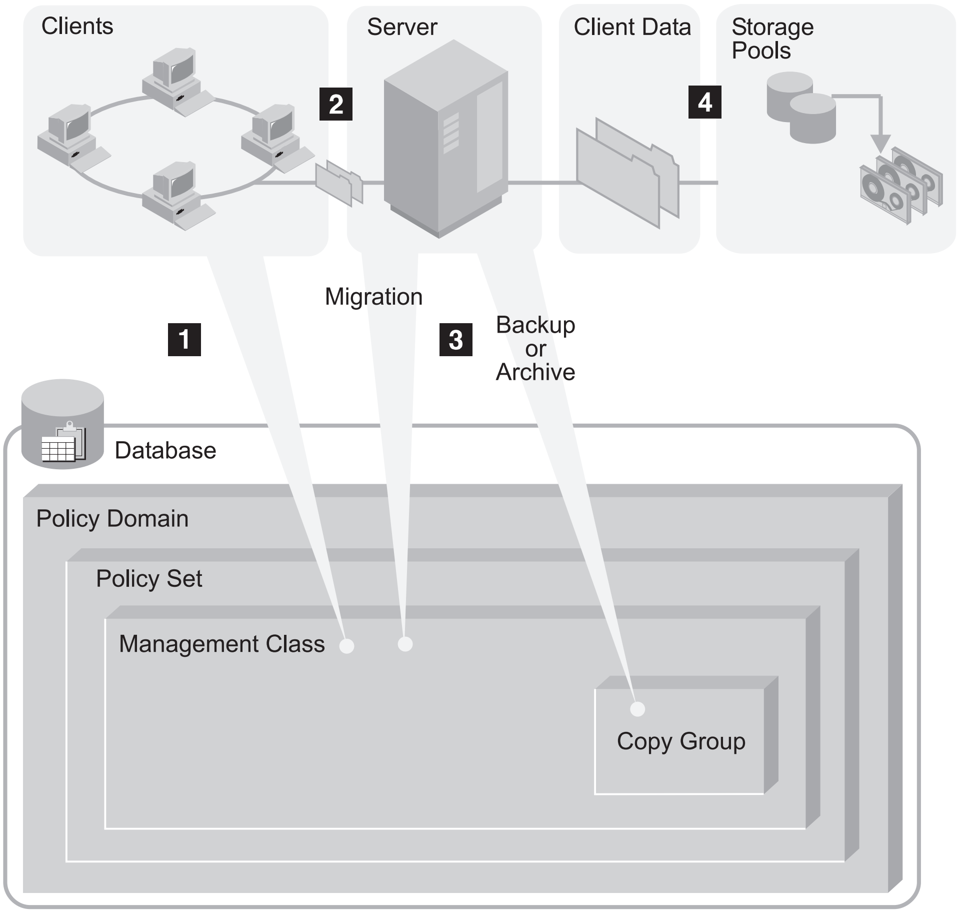


Figure 1. How Tivoli Storage Manager Controls Backup, Archive, and Migration Processes

The steps in the process are as follows:

1 A client initiates a backup, archive, or migration operation. The file involved in the operation is bound to a management class. The management class is either the default or one specified for the file in client options (the client's include-exclude list).

2 If the file is a candidate for backup, archive, or migration based on information in the management class, the client sends the file and file information to the server.

3 The server checks the management class that is bound to the file to determine the *destination*, the name of the Tivoli Storage Manager storage pool where the server initially stores the file. For backed-up and archived files, destinations are assigned in the backup and archive copy groups, which are within management classes. For space-managed files, destinations are assigned in the management class itself.

The storage pool can be a group of disk volumes, tape volumes, or optical volumes.

4 The server stores the file in the storage pool that is identified as the storage destination.

The Tivoli Storage Manager server saves information in its database about each file that it backs up, archives, or migrates.

If you set up server storage in a hierarchy, Tivoli Storage Manager can later migrate the file to a storage pool different from the one where the file was initially stored. For example, you may want to set up server storage so that Tivoli Storage Manager migrates files from a disk storage pool to tape volumes in a tape storage pool.

Files remain in server storage until they expire and expiration processing occurs, or until they are deleted from server storage. A file expires because of criteria that are set in policy. For example, the criteria include the number of versions allowed for a file and the number of days that have elapsed since a file was deleted from the client's file system.

For information on assigning storage destinations in copy groups and management classes, and on binding management classes to client files, see "Chapter 12. Implementing Policies for Client Data" on page 269.

For information on managing the database, see "Chapter 18. Managing the Database and Recovery Log" on page 383.

For information about storage pools and storage pool volumes, see "Chapter 9. Managing Storage Pools and Volumes" on page 155.

Options for Data Protection

Tivoli Storage Manager provides a variety of backup and archive operations, allowing you to select the right protection for the situation. Table 1 shows some examples of the protection options.

Table 1. Examples of Meeting Your Goals with Tivoli Storage Manager

For this goal...	Do this...
Back up files that are on a user's workstation, and have the ability to restore individual files.	Use the backup-archive client to perform incremental backups or selective backups.
Back up a file server, and have the ability to restore individual files.	Use the backup-archive client to perform incremental backups or selective backups. If the file server is a network-attached storage file server that is supported, you can use Tivoli Data Protection for NDMP to perform image backups.
Make restore media portable, or make restores easier to perform remotely.	Use the backup-archive client to perform incremental backups, and then generate backup sets by using the Tivoli Storage Manager server.

Table 1. Examples of Meeting Your Goals with Tivoli Storage Manager (continued)

For this goal...	Do this...
Provide the ability to more easily restore the entire contents of a single logical volume, instead of restoring individual files.	Use the backup-archive client to perform logical volume backups (also called image backups).
Set up records retention to meet legal or other long-term storage needs.	Use the backup-archive client to occasionally perform archiving. To ensure that the archiving occurs at the required intervals, use central scheduling.
Create an archive for a backup-archive client, from data that is already stored for backup.	Use the backup-archive client to perform incremental backups, and then generate a backup set by using the Tivoli Storage Manager server. This is also called *instant archive*.
Provide the ability to restore data to a point in time.	Use the backup-archive client to regularly perform incremental backups (either manually or automatically through schedules). Then do one of the following: • Set up policy to ensure that data is preserved in server storage long enough to provide the required service level. See "Setting Policy to Enable Point-in-Time Restore for Clients" on page 308 for details. • Create backup sets for the backup-archive client on a regular basis. Set the retention time to provide the required service level. See "Creating and Using Client Backup Sets" on page 316 for details.
Save a set of files and directories before making significant changes to them.	Use the backup-archive client to archive the set of files and directories. If this kind of protection is needed regularly, consider creating backup sets from backup data already stored for the client. Using backup sets instead of frequent archive operations can reduce the amount of metadata that must be stored in the server's database.
Back up data for an application that runs continuously, such as a database application (for example, DB2® or Oracle) or a mail application (Lotus® Domino™).	Use the appropriate application client. For example, use Tivoli Data Protection for Lotus Domino.
Exploit disk hardware capable of data snapshots.	Use the appropriate component that supports the hardware, such as Tivoli Data Protection for IBM Enterprise Storage Server™.
Make backups transparent to end users.	Use the backup-archive client with centrally scheduled backups that run during off-shift hours. Monitor the schedule results.
Reduce the load on the LAN by moving backup data over your SAN.	Use LAN-free data movement or, for supported network-attached storage (NAS) file servers, use Tivoli Data Protection for NDMP.

Schedule the backups of client data to help enforce the data management policy that you establish. If you schedule the backups, rather than rely on the clients to perform the backups, the policy that you establish is followed more consistently. See “Chapter 14. Scheduling Operations for Client Nodes” on page 325.

The standard backup method that Tivoli Storage Manager uses is called *progressive incremental backup*. It is a unique and efficient method for backup. See “Progressive Incremental Backup Compared with Other Backup Types” on page 13.

Table 2 on page 10 summarizes the client operations that are available. In all cases, the server tracks the location of the backup data in its database. Policy that you set determines how the backup data is managed.

Table 2. Summary of Client Operations

Type of operation	Description	Usage	Restore options	For more information
Progressive incremental backup	The standard method of backup used by Tivoli Storage Manager. After the first, full backup of a client system, incremental backups are done. Incremental backup by date is also available. No additional full backups of a client are required after the first backup.	Helps ensure complete, effective, policy-based backup of data. Eliminates the need to retransmit backup data that has not been changed during successive backup operations.	The user can restore just the version of the file that is needed. Tivoli Storage Manager does *not* need to restore a base file followed by incremental backups. This means reduced time and fewer tape mounts, as well as less data transmitted over the network.	See “Incremental Backup” on page 284 and *Backup-Archive Installation and User’s Guide*.
Selective backup	Backup of files that are selected by the user, regardless of whether the files have changed since the last backup.	Allows users to protect a subset of their data independent of the normal incremental backup process.	The user can restore just the version of the file that is needed. Tivoli Storage Manager does *not* need to restore a base file followed by incremental backups. This means reduced time and fewer tape mounts, as well as less data transmitted over the network.	See “Selective Backup” on page 286 and *Backup-Archive Installation and User’s Guide*.
Adaptive subfile backup	A backup method that backs up only the *parts* of a file that have changed since the last backup. The server stores the base file (the complete initial backup of the file) and subsequent subfiles (the changed parts) that depend on the base file. The process works with either the standard progressive incremental backup or with selective backup. Applicable to clients on Windows® systems.	Maintains backups of data while minimizing connect time and data transmission for the backup of mobile and remote users.	The base file plus a maximum of one subfile is restored to the client.	See “Enabling Clients to Use Subfile Backup” on page 322 and *Backup-Archive Installation and User’s Guide*.

Table 2. Summary of Client Operations (continued)

Type of operation	Description	Usage	Restore options	For more information
Journal-based backup	Aids all types of backups (progressive incremental backup, selective backup, adaptive subfile backup) by basing the backups on a list of changed files. The list is maintained on the client by the journal engine service of Tivoli Storage Manager.	Reduces the amount of time required for backup. The files eligible for backup are known before the backup operation begins. Applicable to clients on Windows NT® and Windows 2000 systems.	Journal-based backup has no effect on how files are restored; this depends on the type of backup performed.	See *Backup-Archive Installation and User's Guide*.
Image backup	Full volume backup. Nondisruptive, on-line backup is possible for Windows 2000 clients by using the Tivoli Storage Manager snapshot function.	Allows backup of an entire file system or raw volume as a single object. Can be selected by backup-archive clients on UNIX® and Windows systems.	The entire image is restored.	See "Policy for Logical Volume Backups" on page 304 and *Backup-Archive Installation and User's Guide*.
Image backup with differential backups	Full volume backup, which can be followed by subsequent differential backups.	Used only for the image backups of NAS file servers, performed by using Tivoli Data Protection for NDMP.	The full image backup plus a maximum of one differential backup are restored.	See "Chapter 6. Setting Up Tivoli Data Protection for NDMP" on page 97.
Backup using hardware snapshot capabilities	A method of backup that exploits the capabilities of IBM Enterprise Storage Server FlashCopy™ and EMC TimeFinder to make copies of volumes used by database servers. The Tivoli Data Protection for applications and databases then uses the volume copies to back up the database volumes.	Implements high-efficiency backup and recovery of business-critical applications while virtually eliminating backup-related downtime or user disruption on the database server.	See the referenced publications for details.	See *Tivoli Data Protection for IBM ESS for Oracle Databases DB2 UDB Installation and User's Guide*, *Tivoli Data Protection for IBM ESS for R/3 Installation and User's Guide*, and *Tivoli Data Protection for EMC Symmetrix for R/3 Installation and User's Guide*.

Table 2. Summary of Client Operations (continued)

Type of operation	Description	Usage	Restore options	For more information
Archive	The process creates a copy of files and stores them for a specific time.	Use for maintaining copies of vital records for legal or historical purposes. **Note:** If you need to frequently create archives for the same data, consider using instant archive (backup sets) instead. Frequent archive operations can create a large amount of metadata in the server database resulting in increased database growth and decreased performance for server operations such as expiration. Frequently, you can achieve the same objectives with incremental backup or backup sets. Although the archive function is a powerful way to store inactive data with fixed retention, it should not be used on a frequent and large scale basis as the primary backup method.	The selected version of the file is retrieved on request.	See "Archive" on page 287 and *Backup-Archive Installation and User's Guide*.
Instant archive	The process creates a backup set of the most recent versions of the files for the client, using files already in server storage from earlier backup operations.	Use when portability of the recovery media or rapid recovery of a backup-archive client is important. Also use for efficient archiving.	The files are restored directly from the backup set. The backup set resides on media that can be mounted on the client system, such as a CD, a tape drive, or a file system. The Tivoli Storage Manager server does not have to be contacted for the restore process, so the process does not use the network or the server.	See "Creating and Using Client Backup Sets" on page 316.

Progressive Incremental Backup Compared with Other Backup Types

Tivoli Storage Manager has a unique, efficient method for its standard backups, as well as a number of other methods that are summarized in Table 2 on page 10. The standard method that Tivoli Storage Manager uses is *progressive incremental backup*.

The terms *differential* and *incremental* are often used to describe backups. The terms usually have the following meanings:

- A differential backup backs up files that have changed since the last full backup.
  - – If a file changes after the full backup, the changed file is backed up again by *every* subsequent differential backup.
  - – All files are backed up at the next full backup.
- An incremental backup backs up only files that have changed since the last backup, whether that backup was a full backup or another incremental backup.
  - – If a file changes after the full backup, the changed file is backed up *only* by the next incremental backup, not by all subsequent incremental backups.
  - – If a file has not changed since the last backup, the file is not backed up.

Tivoli Storage Manager takes incremental backup one step further. After the initial full backup of a client, no additional full backups are necessary because the server, using its database, keeps track of whether files need to be backed up. Only files that change are backed up, and then entire files are backed up, so that the server does not need to reference base versions of the files. This means savings in resources, including the network and storage.

If you choose, you can force full backup by using the selective backup function of a client in addition to the incremental backup function. You can also choose to use adaptive subfile backup, in which the server stores the base file (the complete initial backup of the file) and subsequent subfiles (the changed parts) that depend on the base file.

Additional Protection: Storage Pool and Server Database Backups

Built into the server are additional levels of protection for client data:

- You can back up storage pools. The data is backed up to copy storage pools, which the server can automatically access if needed to retrieve a file. See “Storage Pool Protection: An Overview” on page 500.
- You can back up the server’s database. The database is key to the server’s ability to track client data in server storage. See “Database and Recovery Log Protection: An Overview” on page 501.

These backups can become part of a disaster recovery plan, created automatically by Tivoli Disaster Recovery Manager. See “Chapter 23. Using Tivoli Disaster Recovery Manager” on page 545.

How Data Moves to Server Storage

The Tivoli Storage Manager client traditionally sends its data to the server over the LAN. The server then transfers the data to a device that is attached to the server. With the advent of SAN and network-attached storage, however, Tivoli Storage Manager offers options that enable you to minimize use of the LAN and the use of the computing resources of both the client and the server.

LAN-free data movement allows storage agents that are installed on client nodes to move data without sending the data over the LAN to the server. See "LAN-Free Data Movement" on page 36.

For network-attached storage, use Tivoli Data Protection for NDMP to avoid data movement over the LAN. See "Tivoli Data Protection for NDMP Backups" on page 35.

Consolidating Backed-up Data for Clients

By grouping the backed-up data for a client, you can minimize the number of media mounts required for client recovery. The server offers you methods for doing this:

Collocation

The server can keep each client's files on a minimal number of volumes within a storage pool. Because client files are consolidated, restoring collocated files requires fewer media mounts. However, backing up files from different clients requires more mounts.

You can have the server collocate client data when the data is initially stored in server storage. If you have a storage hierarchy, you can also have the data collocated when the server migrates the data from the initial storage pool to the next storage pool in the storage hierarchy.

Another choice you have is the level of collocation. You can collocate by client or by file space per client. Your selection depends on the size of the file spaces being stored and the restore requirements.

See "Keeping a Client's Files Together: Collocation" on page 184.

Backup set creation

You can generate a backup set for each backup-archive client. A backup set contains all active backed-up files that currently exist for that client in server storage. The process is also called instant archive.

The backup set is portable and is retained for the time that you specify. Creation of the backup set consumes more media because it is a copy in addition to the backups that are already stored.

See "Creating and Using Client Backup Sets" on page 316.

Moving data for a client node

You can consolidate data for a client node by moving the data within server storage. You can move it to a different storage pool, or to other volumes in the same storage pool.

See "Moving Data by Node" on page 215.

How the Server Manages Storage

Through the server, you manage its storage — the devices and media used to store client data. The server integrates the management of storage with the policies that you define for managing client data.

Tivoli Storage Manager Device Support

Tivoli Storage Manager supports the use of a variety of devices for server storage. Tivoli Storage Manager can use direct-attached storage as well as network-attached storage. A current list is available at the Tivoli Storage Manager Web site:

http://www.tivoli.com/support/storage_mgr/tivolimain.html

Tivoli Storage Manager represents physical storage devices and media with the following administrator-defined objects:

Library
A library is one or more drives (and possibly robotic devices) with similar media mounting requirements.

Drive
Each drive represents a drive mechanism in a tape or optical device.

Data mover
A data mover represents a device that accepts requests from Tivoli Storage Manager to transfer data on behalf of the server. Data movers transfer data between storage devices.

Path
A path represents how a source accesses a destination. For example, the source can be a server, and the destination can be a tape drive. A path defines the one-to-one relationship between a source and a destination. Data may flow from the source to the destination, and back.

Device class
Each device is associated with a device class that specifies the device type and how the device manages its media.

Storage pools and volumes
A storage pool is a named collection of volumes that have the same media type. A storage pool is associated with a device class. For example, an 8mm tape storage pool contains only 8mm tape volumes. A storage pool volume is associated with a specific storage pool.

For details about device concepts, see “Chapter 2. Introducing Storage Devices” on page 29.

Migrating Data through the Storage Hierarchy

You can organize the server’s storage pools into one or more hierarchical structures. This storage hierarchy allows flexibility in a number of ways. For example, you can set policy to have clients send their backup data to disks for faster backup operations, then later have the server automatically migrate the data to tape.

See “Overview: The Storage Pool Hierarchy” on page 170.

Removing Expired Data

Policy that you define controls when client data automatically expires from the Tivoli Storage Manager server. The expiration process is how the server implements the policy.

For example, you have a backup policy that specifies that three versions of a file be kept. File A is created on the client, and backed up. Over time, the user changes file A, and three versions of the file are backed up to the server. Then the user changes file A again. When the next incremental backup occurs, a fourth version of file A is stored, and the oldest of the four versions is eligible for expiration.

To remove data that is eligible for expiration, a server expiration process marks data as expired and deletes metadata for the expired data from the database. The space occupied by the expired data is then available for new data.

You control the frequency of the expiration process by using a server option, or you can start the expiration processing by command or scheduled command.

See “Running Expiration Processing to Delete Expired Files” on page 302.

Reclaiming Media for Reuse

As server policies automatically expire data, the media where the data is stored accumulates unused space. Other products might require you to implement a tape rotation scheme that allows you to reuse the media only when all data on the media has expired. The Tivoli Storage Manager server implements a different process, called reclamation, that allows you to reuse media without traditional tape rotation.

Reclamation is a server process that automatically defragments media by consolidating unexpired data onto other media when the free space on media reaches a defined level. The reclaimed media can then be used again by the server. Reclaiming media allows the automated circulation of media through the storage management process. Use of reclamation can help minimize the number of media that you need to have available.

Configuring and Maintaining the Server

The server comes with many defaults set so that clients can begin using its services immediately. The amount and importance of the data protected by Tivoli Storage Manager, your business process requirements, and other factors make it likely that you need to adjust and customize the server’s behavior. Your changing storage needs and client requirements can mean on-going configuration changes and monitoring.

The server’s capabilities are extensively described in this guide. To get an introduction to the tasks available to an administrator of Tivoli Storage Manager, read the following sections:

“Interfaces to Tivoli Storage Manager”
“Configuring and Managing Server Storage” on page 17
“Managing Client Operations” on page 20
“Maintaining the Server” on page 23
“Protecting the Server” on page 26

Interfaces to Tivoli Storage Manager

Tivoli Storage Manager has the following types of interfaces:

- Graphical user interfaces.

  For the clients, there are graphical user interfaces for the backup-archive client and the space manager client (if installed, on supported operating systems). For information about using the interfaces, see the online information or see *Quick Start*.
- Web interfaces for server administration and for the backup-archive client.

  The administrative Web interface allows you to access Tivoli Storage Manager server functions from any workstation with a Web browser that has the appropriate support for Java®. The interface also allows Web access to the command line. See *Quick Start* for information about the administrative Web interface.

The Web backup-archive client (Web client) allows an authorized user to remotely access a client to run backup, archive, restore, and retrieve processes. The Web browser must have the appropriate support for Java. See *Backup-Archive Installation and User's Guide* for requirements.

- The command-line interface.

  For information about using the command-line interface of the administrative client, see *Administrator's Reference*. For information about using the command-line interface of the backup-archive client and other clients, see *Backup-Archive Installation and User's Guide* for that client.
- The application program interface.

  For more information, see *Tivoli Storage Manager Using the Application Program Interface*.
- Access to information in the server's database via standard SQL SELECT statements. For reporting purposes, the Tivoli Storage Manager product also provides an ODBC driver. The driver allows you to use a spreadsheet or database program to query the server database for information. See "Using SQL to Query the Tivoli Storage Manager Database" on page 406.

Customizing the Server with the Server Options File

Server options allow you to customize the server and its operations. Some examples of what these options affect are:

- Server communications
- Storage
- Database and recovery log operations
- Client transaction performance

Server options are in the server options file. Some options can be changed and made active immediately by using the command, SETOPT. Most server options are changed by editing the server options file and then halting and restarting the server to make the changes active. In this book, specific server options are discussed where they are applicable. See *Administrator's Reference* for details about the server options file and reference information for all server options.

Configuring and Managing Server Storage

Configuring and managing storage for efficiency and capacity are important tasks for an administrator.

The server uses its storage for the data it manages for clients. The storage can be a combination of devices:

- Disk
- Tape drives that are either manually operated or automated
- Optical drives
- Other drives that use removable media

The devices can be locally attached, or accessible through a SAN. Key decisions in configuring and managing the storage include:

- Selecting the devices and media that will form the server storage, including library devices that will be shared among Tivoli Storage Manager servers
- Designing the storage hierarchy for efficient backups and optimal storage usage
- Using product features that allow the server to provide services to clients while minimizing traffic on the communications network:

  - LAN-free data movement
  - Data movement using NDMP to protect data on network-attached storage (NAS) file servers
- Using the Tivoli Storage Manager product to help you to manage the drives and media, or using an external media manager to do the management outside of the Tivoli Storage Manager product.

For an introduction to key storage concepts, see "Chapter 2. Introducing Storage Devices" on page 29.

Using Hard Disk Devices with Tivoli Storage Manager

Hard disk devices can be used with Tivoli Storage Manager for two purposes:

- Storage of the database and recovery log
- Storage of client data that is backed up, archived, or migrated from client nodes

The server can store data on hard disk by using random access volumes (device type of DISK) or sequential access volumes (device type of FILE).

The Tivoli Storage Manager product allows you to exploit disk storage in ways that other products do not. You can have multiple client nodes back up to the same disk storage pool at the same time, and still keep the data for the different client nodes separate. Other products also allow you to back up different systems at the same time, but only by interleaving the data for the systems, leading to slower restore processes.

If you have enough disk storage space, data can remain on disk permanently or temporarily, depending on the amount of disk storage space that you have. Restore process performance from disk can be very fast compared to tape.

You can have the server later move the data from disk to tape; this is called migration through the storage hierarchy. Other advantages to this later move to tape include:

- Ability to collocate data for clients as the data is moved to tape
- Streaming operation of tape drives, leading to better tape drive performance
- More efficient use of tape drives by spreading out the times when the drives are in use

For information about setting up storage pools on disk devices, see "Chapter 3. Using Magnetic Disk Devices" on page 51. For information about setting up a storage hierarchy, see "Overview: The Storage Pool Hierarchy" on page 170.

Using Removable Media Devices with Tivoli Storage Manager

Removable media devices can be used with Tivoli Storage Manager for the following purposes:

- Storage of client data that is backed up, archived, or migrated from client nodes
- Storage of database backups
- The exporting of data, to move the data to another server

Attaching and Configuring Devices: For information about configuring your removable media devices, see "Chapter 4. Attaching Devices to the Server System" on page 57 and "Chapter 5. Configuring Storage Devices" on page 67.

Classifying Devices by Device Type: A device class represents a set of storage devices with similar availability, performance, and storage characteristics. You must define device classes for the drives available to the Tivoli Storage Manager server. You specify a device class when you define a storage pool so that the storage pool is associated with drives.

For more information about defining device classes, see “Chapter 8. Defining Device Classes” on page 141.

Managing Removable Media Operations: Routine removable media operations including the following:

- Preparing media for use
- Controlling media reuse
- Ensuring that sufficient media are available
- Mounting volumes in response to server requests, for manually operated drives
- Managing libraries and drives

For information about removable media operations, see “Chapter 7. Managing Removable Media Operations” on page 113.

Managing Storage Pools and Volumes

Backed-up, archived, and space-managed files are stored in groups of volumes that are called storage pools. Because each storage pool is assigned to a device class, you can logically group your storage devices to meet your storage management needs.

You can establish a hierarchy of storage pools. The hierarchy may be based on the speed or the cost of the devices associated with the pools. Tivoli Storage Manager migrates client files through this hierarchy to ensure the most efficient use of a server’s storage devices.

The following are other examples of what you can control for a storage pool:

Collocation
The server can keep each client’s files on a minimal number of volumes within a storage pool. Because client files are consolidated, restoring collocated files requires fewer media mounts. However, backing up files from different clients requires more mounts.

Reclamation
Files on sequential access volumes may expire, move, or be deleted. The reclamation process consolidates the active, unexpired data on many volumes onto fewer volumes. The original volumes can then be reused for new data, making more efficient use of media.

Storage pool backup
The data on primary storage pools can be backed up to copy storage pools for disaster recovery purposes. Backup to copy storage pools can occur simultaneously as client data is written to the primary storage pool.

Cache
When the server migrates files from disk storage pools, duplicate copies of the files can remain in cache (disk storage) for faster retrieval. Cached files are deleted only when space is needed. However, client backup operations that use the disk storage pool may have poorer performance.

You manage storage volumes by defining, updating, and deleting volumes, and by monitoring the use of server storage. You can also move files within and across storage pools to optimize the use of server storage.

For more information about storage pools and volumes and taking advantage of storage pool features, see “Chapter 9. Managing Storage Pools and Volumes” on page 155.

Using HACMP for Server Availability

IBM High Availability Cluster Multi-Processing for AIX (HACMP) detects system failures and manages failover to a recovery processor with a minimal loss of end-user time. You can set up a Tivoli Storage Manager server on a system in an HACMP cluster so that, if the system fails, the Tivoli Storage Manager server will be brought back up on another system in the cluster. In both failover and fallback, it appears that the Tivoli Storage Manager server has crashed or halted and was then restarted. Any transactions that were in progress at the time of the failover or fallback are rolled back, and all completed transactions are still complete. Tivoli Storage Manager clients see this as a communications failure and try to reestablish their connections. See *Quick Start* for details.

Managing Client Operations

Because the key task of the server is to provide services to clients, many of the server administrator’s tasks deal with client operations. Tasks include the following:

- Registering clients and customizing client operations
- Ensuring that client operations meet security requirements
- Providing required levels of service by customizing policies
- Automating protection by using schedules

Managing Client Nodes

A very basic administrative task is adding client nodes, giving the systems that the nodes represent access to the services and resources of the Tivoli Storage Manager server. The Tivoli Storage Manager server supports a variety of client nodes. You can register the following types of clients and servers as client nodes:

- Tivoli Storage Manager backup-archive client
- Tivoli Data Protection application clients
- Tivoli Space Manager client (called space manager client or HSM client)
- A NAS file server that uses Tivoli Data Protection for NDMP for backup
- Tivoli Storage Manager source server (registered as a node on a target server)

When you register clients, you have choices to make about the following:

- Whether the client should compress files before sending them to the server for backup
- Whether the client node ID has the authority to delete its files from server storage
- Whether an administrator ID that matches the client ID is created, for remote client operations

Other important tasks include the following:

Controlling client options from the server

Client options on client systems allow users to customize backup, archive, and space management operations, as well as schedules for these operations. On most client systems, the options are in a file called *dsm.opt*. In some cases, you may need or want to provide the clients with options to use. To help users get started, or to control what users back up, you can define sets of client options for clients to use. Client options sets are defined in the server database and are used by the clients that you designate.

Among the options that can be in a client option set are the include and exclude options. These options control which files are considered for the client operations.

For more information, see “Chapter 11. Managing Client Nodes” on page 235.

Allowing subfile backups

For mobile and remote users, you want to minimize the data sent over the network, as well as the time that they are connected to the network. You can set the server to allow a client node to back up changed portions of files that have been previously backed up, rather than entire files. The portion of the file that is backed up is called a *subfile*.

For more information, see “Chapter 13. Managing Data for Client Nodes” on page 315.

Creating backup sets for client nodes

You can perform an instant archive for a client by creating a backup set. A backup set copies a client node’s active, backed-up files from server storage onto sequential media. If the sequential media can be read by a device available to the client system, you can restore the backup set directly to the client system without using the network. The server tracks backup sets that you create and retains the backup sets for the time you specify.

For more information, see “Chapter 13. Managing Data for Client Nodes” on page 315 .

For more information, see “Chapter 10. Adding Client Nodes” on page 227 and “Chapter 11. Managing Client Nodes” on page 235.

Managing Security

Tivoli Storage Manager includes security features for user registration and passwords.

Registration for clients can be closed or open. With closed registration, a user with administrator authority must register all clients. With open registration, clients can register themselves at first contact with the server. See “Registering Nodes with the Server” on page 228.

You can ensure that only authorized administrators and client nodes are communicating with the server by requiring the use of passwords. You can also set the following requirements for passwords:

- Number of characters in a password.
- Expiration time.
- A limit on the number of consecutive, invalid password attempts. When the client exceeds the limit, Tivoli Storage Manager locks the client node from access to the server.

See “Managing Passwords” on page 266.

You can control the authority of administrators. An organization may name a single administrator or may distribute the workload among a number of administrators and grant them different levels of authority. For details, see “Managing Levels of Administrative Authority” on page 264.

Implementing Policies for Client Data

As the administrator, you define the rules for client backup, archive, and migration operations, based on user or business requirements. The rules are called *policies*. Policies identify:

- The criteria for backup, archive, and migration of client data
- Where the client data is initially stored
- How the data is managed by the server (how many backup versions are kept, for how long)

In Tivoli Storage Manager, you define policies by defining policy domains, policy sets, management classes, and backup and archive copy groups. When you install Tivoli Storage Manager, you have a default policy that consists of a single policy domain named STANDARD.

The default policy provides basic backup protection for end-user workstations. To provide different levels of service for different clients, you can add to the default policy or create new policy. For example, because of business needs, file servers are likely to require a policy different from policy for end users. Protecting data for applications such as Lotus Domino also may require a unique policy.

For more information about the default policy and establishing and managing new policies, see "Chapter 12. Implementing Policies for Client Data" on page 269.

Scheduling Client Operations

Scheduling client operations can mean better protection for data, because operations can occur consistently without user intervention. Scheduling also can mean better utilization of resources such as the network. Client backups that are scheduled at times of lower usage can minimize the impact on user operations on a network.

You can automate operations for clients by using schedules. Tivoli Storage Manager provides a central scheduling facility. You can also use operating system utilities or other scheduling tools to schedule Tivoli Storage Manager operations.

With Tivoli Storage Manager schedules, you can perform the operations for a client immediately or schedule the operations to occur at regular intervals.

The key objects that interact are:

Include-exclude options on each client

The include-exclude options determines which files are backed up, archived, or space-managed, and determines management classes, encryption, and type of backup for files.

The client can specify a management class for a file or group of files, or can use the default management class for the policy domain. The client specifies a management class by using an INCLUDE option in the client's include-exclude list or file. You can have central control of client options such as INCLUDE and EXCLUDE by defining client option sets on the server. When you register a client, you can specify a client option set for that client to use. See "Modifying Client Option Files" on page 253 for details.

Association defined between client and schedule

Associations determine which schedules are run for a client.

Clients are assigned to a policy domain when they are registered. To automate client operations, you define schedules for a domain. Then you define associations between schedules and clients in the same domain.

Schedule
The schedule determines when a client operation automatically occurs.

Schedules that can automate client operations are associated with a policy domain.

The scheduled client operations are called *events*. The Tivoli Storage Manager server stores information about events in its database. For example, you can query the server to determine which scheduled events completed successfully and which failed.

Management class
The management class determines where client files are initially stored and how they are managed.

The management class contains information that determines how Tivoli Storage Manager handles files that clients backup, archive, or migrate. For example, the management class contains the backup copy group and the archive copy group. Each copy group points to a *destination*, a storage pool where files are first stored when they are backed up or archived.

For a schedule to work on a particular client, the client machine must be turned on. The client must either be running the client scheduler or must allow the client acceptor daemon to start the scheduler when needed.

To learn more:

- For how to set up policy domains and management classes, see “Chapter 12. Implementing Policies for Client Data” on page 269.
- For how to automate client operations, see “Chapter 14. Scheduling Operations for Client Nodes” on page 325.
- For how to set up an include-exclude list for clients, see “Getting Users Started” on page 272.
- For how to run the scheduler on client systems, see the user guides for the clients.

After you have created schedules, you manage and coordinate those schedules. Your tasks include the following:

- Verify that the schedule ran successfully.
- Determine how long Tivoli Storage Manager retains event records in the database.
- Balance the workload on the server so that all scheduled operations complete.

For more information about these tasks, see “Chapter 15. Managing Schedules for Client Nodes” on page 333.

Maintaining the Server

To keep the server running well, you have access to these tasks:

- Managing server operations, such as controlling client access to the server
- Automating repetitive administrative tasks
- Monitoring and adjusting space for the database and the recovery log
- Monitoring the status of the server, server storage, and clients

If you manage more than one server, you can ensure that the multiple servers are consistently managed by using the enterprise management functions of Tivoli Storage Manager. You can set up one server as the configuration manager and have other servers obtain configuration information from it.

Managing Server Operations

There are a variety of tasks associated with managing server operations:

- Start and stop the server.
- Allow and suspend client sessions with the server.
- Query, cancel, and preempt server processes such as backing up the server database.
- Customize server options.

Other tasks that are needed less frequently include:

- Maintain compliance with the license agreement.
- Move the server.

See "Licensing Tivoli Storage Manager" on page 349. For suggestions about the day-to-day tasks required to administer the server, see "Chapter 16. Managing Server Operations" on page 349.

Automating Server Operations

Repetitive, manual tasks associated with managing the server can be automated through Tivoli Storage Manager schedules and scripts. Using schedules and scripts can minimize the daily tasks for administrators.

You can define schedules for the automatic processing of most administrative commands. For example, a schedule can run the command to back up the server's database every day.

Tivoli Storage Manager server scripts allow you to combine administrative commands with return code checking and processing. The server comes with scripts that you can use to do routine tasks, or you can define your own. The scripts typically combine several administrative commands with return code checking, or run a complex SQL SELECT command. Scripts can also be scheduled.

For more information about automating Tivoli Storage Manager operations, see "Chapter 17. Automating Server Operations" on page 367.

Managing the Database and Recovery Log

The Tivoli Storage Manager database contains information about registered client nodes, policies, schedules, and the client data in storage pools (metadata). The information about the client data, also called *metadata*, includes the file name, file size, file owner, management class, copy group, and location of the file in server storage. The database is key to the operation of the server.

The server records changes made to the database (database transactions) in its recovery log. The recovery log is used to maintain the database in a transactionally consistent state, and to maintain consistency across server start-up operations.

You can tune database and recovery log performance automatically or manually. You can set up triggers so that additional space is automatically added to the database and recovery log as needed.

For more information about the Tivoli Storage Manager database and recovery log and about the tasks associated with them, see "Chapter 18. Managing the Database and Recovery Log" on page 383.

Monitoring the Tivoli Storage Manager Server

Tivoli Storage Manager provides you with many sources of information about server and client status and activity, the state of the server's database and storage, and resource usage. By monitoring selected information, you can provide reliable services to users while making the best use of available resources. Daily checks of some indicators are suggested.

You can use Tivoli Storage Manager queries and SQL queries to get information about the server. An ODBC interface is available.

You can set up automatic logging of information about Tivoli Storage Manager clients and server events.

See the following sections for more information about these tasks.

- "Chapter 19. Monitoring the Tivoli Storage Manager Server" on page 401
- "Using SQL to Query the Tivoli Storage Manager Database" on page 406
- "Logging Tivoli Storage Manager Events to Receivers" on page 413
- "Daily Monitoring Scenario" on page 428

Working with a Network of Tivoli Storage Manager Servers

You may have a number of Tivoli Storage Manager servers in your network, at the same or different locations. Some examples are:

- Your users are scattered across many locations, so you have located Tivoli Storage Manager servers close to the users to manage network bandwidth limitations.
- You have set up multiple servers to provide services to different organizations at one location.
- You have multiple servers on your network to make disaster recovery easier.

Servers connected to a network can be centrally managed. Tivoli Storage Manager provides functions to help you configure, manage, and monitor the servers. An administrator working at one Tivoli Storage Manager server can work with servers at other locations around the world.

When you have a network of Tivoli Storage Manager servers, you can simplify configuration and management of the servers by using Enterprise Administration functions. You can do the following:

- Designate one server as a configuration manager that distributes configuration information such as policy to other servers. See "Setting Up an Enterprise Configuration" on page 443.
- Route commands to multiple servers while logged on to one server. See "Routing Commands" on page 465.
- Log events such as error messages to one server. This allows you to monitor many servers and clients from a single server. See "Enterprise Event Logging: Logging Events to Another Server" on page 423.
- Store data for one Tivoli Storage Manager server in the storage of another Tivoli Storage Manager server. The storage is called server-to-server virtual volumes. See "Using Virtual Volumes to Store Data on Another Server" on page 469 for details.

- Share an automated library among Tivoli Storage Manager servers. See "Multiple Tivoli Storage Manager Servers Sharing Libraries" on page 35.
- Store a recovery plan file for one server on another server, when using Tivoli Disaster Recovery Manager. You can also back up the server database and storage pools to another server. See "Chapter 23. Using Tivoli Disaster Recovery Manager" on page 545 for details.

Exporting and Importing Data

As conditions change, you can move data from one server to another. You can *export* part or all of a server's data so that you can then *import* the data to another server. The export process writes the data to sequential media, such as tape or a file on hard disk. For example, you can balance workload among servers by moving client nodes from one server to another.

For more information about moving data between servers, see "Chapter 21. Exporting and Importing Data" on page 477.

Protecting the Server

Because the server is protecting client data, it is important to protect the server itself.

Tivoli Storage Manager provides a number of ways to protect and recover your server from media failure or from the loss of the Tivoli Storage Manager database or storage pools. Recovery is based on the following preventive measures:

- Mirroring, by which the server maintains one or more copies of the database or the recovery log, allowing the system to continue when one of the mirrored disks fails
- Periodic backup of the database
- Periodic backup of the storage pools
- Audit of storage pools for damaged files, and recovery of damaged files when necessary
- Backup of the device configuration and volume history files
- Validation of the data in storage pools, using cyclic redundancy checking

In addition to taking these actions, you can prepare a disaster recovery plan to guide you through the recovery process by using the Tivoli Disaster Recovery Manager. Tivoli Disaster Recovery Manager is an optional product that assists you in the automatic preparation of a disaster recovery plan. You can use the disaster recovery plan as a guide for disaster recovery as well as for audit purposes to certify the recoverability of the Tivoli Storage Manager server.

The disaster recovery methods of Tivoli Disaster Recovery Manager are based on taking the following measures:

- Sending server backup volumes offsite or to another Tivoli Storage Manager server
- Creating the disaster recovery plan file for the Tivoli Storage Manager server
- Storing client machine information
- Defining and tracking client recovery media

For more information about protecting your server and for details about recovering from a disaster, see "Chapter 22. Protecting and Recovering Your Server" on page 499.

Part 2. Configuring and Managing Server Storage

© Copyright IBM Corp. 1993, 2002

Chapter 2. Introducing Storage Devices

This chapter introduces key concepts that you must be familiar with to work with Tivoli Storage Manager storage devices. It also describes what you will find in the storage device chapters.

How to Use the Server Storage Chapters

If you are new to Tivoli Storage Manager, you should begin by familiarizing yourself with the concepts presented in this chapter. The other chapters in this part of the book will help you to do the following:

Goal	Chapter
Configure and manage magnetic disk devices, which Tivoli Storage Manager uses to store client data, the database, database backups, recovery log, and export data.	"Chapter 3. Using Magnetic Disk Devices" on page 51
Physically attach storage devices to your system and to install and configure the required device drivers.	"Chapter 4. Attaching Devices to the Server System" on page 57
Configure devices to use with Tivoli Storage Manager, and to see detailed scenarios of representative device configurations.	"Chapter 5. Configuring Storage Devices" on page 67
Perform routine operations such as labeling volumes, checking volumes into automated libraries, and maintaining storage volumes and devices.	"Chapter 7. Managing Removable Media Operations" on page 113
Define and manage device classes.	"Chapter 8. Defining Device Classes" on page 141
Understand storage pool and storage volume concepts, and to define and manage storage pools and storage volumes.	"Chapter 9. Managing Storage Pools and Volumes" on page 155

Tivoli Storage Manager Storage Devices

Tivoli Storage Manager devices may be real physical devices, such as disk drives or tape drives, or logical devices, such as files on a disk or storage on another server. See the following sections for details:

- "Disk Devices" on page 30
- "Libraries" on page 30
- "Drives" on page 30

© Copyright IBM Corp. 1993, 2002

- "Files on Disk as Sequential Volumes" on page 31
- "Data Movers" on page 31
- "Sequential Volumes on Another Tivoli Storage Manager Server" on page 32

For a summary, see Table 4 on page 48. For details about specific devices that are supported, visit the Tivoli Storage Manager Web site at this URL http://www.tivoli.com/support/storage_mgr/tivolimain.html.

Disk Devices

Magnetic disk devices are the only random access devices. All disk devices share the same device type and predefined device class: DISK.

Drives

Tivoli Storage Manager drives include tape and optical drives that can stand alone or that can be part of an automated library. Supported removable media drives also include removable files devices such as Iomega Zip drives and Jaz drives.

Libraries

Tivoli Storage Manager supports a variety of library types described in the following sections.

Manual Libraries

In a manual library, an operator mounts the volumes. You can combine drives with different device types, such as DLT and 8mm, in a single manual library.

When the server determines that a volume must be mounted on a drive in a manual library, the server issues mount request messages that prompt an operator to mount the volume. The server sends these messages to the server console and to administrative clients that were started by using the special *mount mode* or *console mode* parameter.

For help on configuring a manual library, see "Chapter 5. Configuring Storage Devices" on page 67. For information on how to monitor mount messages for a manual library, see "Mount Operations for Manual Libraries" on page 127.

SCSI Libraries

A SCSI library, which is controlled through a SCSI interface, is a collection of drives of the same type. In a SCSI library, a robot or other mechanism automatically handles volume mounts and dismounts.

Some examples of this library type are:

- The Exabyte EXB-210 library
- The IBM 3590 tape device, with its Automatic Cartridge Facility (ACF)

Note: The IBM 3494 Tape Library Dataserver, although it has a SCSI interface, is defined as a 349X library type.

For help on configuring a SCSI library, see "Chapter 5. Configuring Storage Devices" on page 67.

349X Libraries

A 349X library is a collection of drives in an IBM 3494 or 3495 Tape Library Dataserver. Volume mounts and demounts are handled automatically by the library. A 349X library has one or more *library management control points* (LMCP) that the server uses to mount and dismount volumes in a drive. Each LMCP provides an independent interface to the robot mechanism in the library.

The drives in a 3494 library can be all of the same type (IBM 3490 or 3590) or a mix of both types. For help on configuring a 349X library, see "Chapter 5. Configuring Storage Devices" on page 67.

External Libraries

An external library is a collection of drives managed by an external media management system that is not part of Tivoli Storage Manager. The server provides an interface that allows external media management systems to operate with the server. The external media management system performs the following functions:

- Volume mounts (specific and scratch)
- Volume dismounts
- Freeing of library volumes (return to scratch)

The external media manager selects the appropriate drive for media access operations. You do not define the drives, check in media, or label the volumes in an external library.

An external library allows flexibility in grouping drives into libraries and storage pools. The library may have one drive, a collection of drives, or even a part of an automated library.

An ACSLS library is a type of external library that is controlled by the StorageTek software, Automated Cartridge System Library Software (ACSLS). The server can act as a client application to the ACSLS software to use the drives.

For a definition of the interface that Tivoli Storage Manager provides to the external media management system, see "Appendix A. External Media Management Interface Description" on page 603.

Data Movers

Data movers are devices that accept requests from Tivoli Storage Manager to transfer data on behalf of the server. Data movers transfer data:

- Between storage devices
- Without using significant Tivoli Storage Manager server or client resources
- Without using significant network resources

For NDMP operations, data movers are NAS file servers. The definition for a NAS data mover contains the network address, authorization, and data formats required for NDMP operations. A data mover enables communication and ensures authority for NDMP operations between the Tivoli Storage Manager server and the NAS file server.

Files on Disk as Sequential Volumes

You can create volumes on server disk space that have the characteristics of sequential access volumes such as tape. Tivoli Storage Manager supports these sequential volumes through the FILE device type. Because FILE is a sequential device type that is on disk, you do not define a library or drive object; only a device class.

FILE volumes can also be useful when transferring data for purposes such as electronic vaulting.

For more information about using, FILE volumes see "Configuring FILE Sequential Volumes on Disk Devices" on page 53.

Sequential Volumes on Another Tivoli Storage Manager Server

You can create volumes that exist as archived files in the storage hierarchy of another Tivoli Storage Manager server. The volumes created are a special type of sequential access volume called a *virtual volume*. Virtual volumes have the characteristics of sequential access volumes such as tape. Tivoli Storage Manager supports virtual volumes through the SERVER device type. You must define a device class and a server that will store the data. No library or drive definition is required.

Virtual volumes are useful for the following purposes:

- Centralization of physical tape resources. You can have one server attached to a large tape library. Other servers can use that library indirectly through a SERVER device class.
- Data-sharing between servers, by using a SERVER device class to export and import data. You do not need to move any physical media from location to location.
- Immediate offsite storage of storage pool backups and database backups, without physically moving media to another location.
- Offsite storage of the Tivoli Disaster Recovery Manager (DRM) recovery plan file.
- Electronic vaulting.

Selecting a Device Configuration

In the conventional local area network (LAN) configuration, one or more tape or optical libraries are associated with a single Tivoli Storage Manager server. In a LAN configuration, client data, electronic mail, terminal connection, application program, and device control information must all be handled by the same network.

A storage area network (SAN) is a dedicated storage network that can improve system performance. On a SAN you can consolidate storage and relieve the distance, scalability, and bandwidth limitations of LANs and wide area networks (WANs). Using Tivoli Storage Manager in a SAN allows the following functions:

- Sharing storage devices among multiple Tivoli Storage Manager servers
- Allowing Tivoli Storage Manager clients, through a storage agent on the client machine, to move data directly to storage devices (LAN-free data movement)

In a SAN you can share storage devices that are supported by the Tivoli Storage Manager device driver. This includes most SCSI devices, but does not include devices that use the GENERICTAPE device type. See "Chapter 4. Attaching Devices to the Server System" on page 57 for device driver setup information.

For information about supported devices and Fibre Channel hardware and configurations, see the following Web sites:

- http://www.tivoli.com/support/storage_mgr/tivolimain.html
- http://www.tivoli.com/support/storage_mgr/san/overview.html

The following sections describe ways that you can configure your storage devices to work with Tivoli Storage Manager:

- “Local Area Network Configuration”
- “Network-Attached Storage”
- Configurations in a storage area network:
  - “Multiple Tivoli Storage Manager Servers Sharing Libraries” on page 35
  - “LAN-Free Data Movement” on page 36

“Planning for Server Storage” on page 38 describes how to evaluate your current storage device environment. It also describes the requirements, advantages, and limitations of all these configurations to help you choose the best set up for your installation.

Local Area Network Configuration

In a local area network configuration, a drive or an automated library can be controlled by only one Tivoli Storage Manager server. Device control information and client backup and restore data flow across the LAN.

Tivoli Storage Manager supports the following categories of libraries:

Manual
A collection of one or more drives that are loaded and unloaded by an operator.

SCSI A SCSI-controlled automated library that includes one or more drives.

349X An IBM 3494 automated library and associated drives.

External
A library that is controlled by a media management program rather than by Tivoli Storage Manager. This category includes StorageTek libraries controlled by Automated Cartridge System Library Software (ACSLS).

For more detailed information, see “Libraries” on page 30.

Network-Attached Storage

Network-attached storage (NAS) file servers are dedicated storage machines whose operating systems are optimized for file-serving functions. NAS file servers typically do not run third-party software. Instead, they interact with programs like Tivoli Storage Manager through industry-standard network protocols, such as NDMP. Tivoli Storage Manager uses Tivoli Data Protection for NDMP to communicate with and provide backup and restore data for NAS file servers.

Tivoli Data Protection for NDMP backs up and restores images of complete file systems. NDMP allows the Tivoli Storage Manager server to control the backup of a NAS file server. The file server transfers the backup data to a drive in a SCSI-attached tape library. The NAS file server can be distant from the Tivoli Storage Manager server.

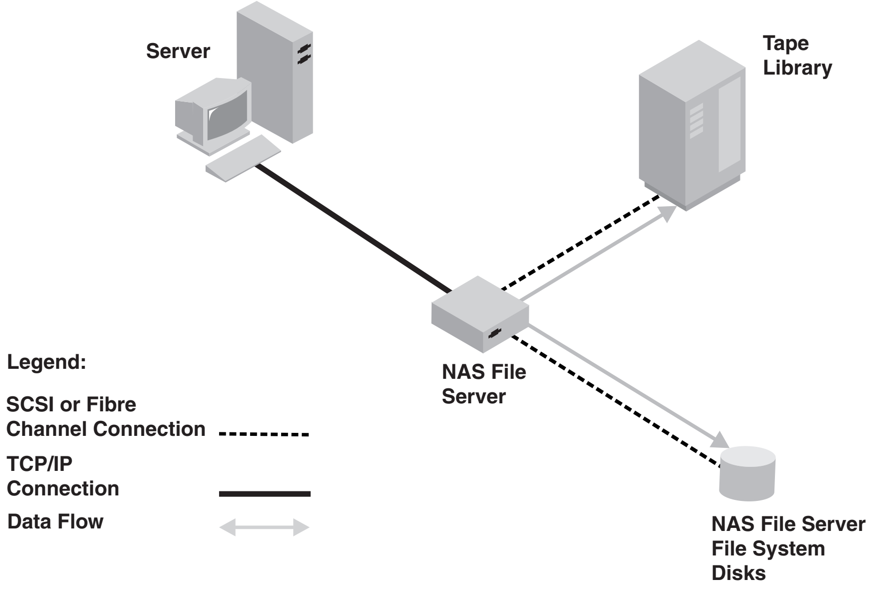


Figure 2. Network- Attached Storage (NAS) Configuration

Tivoli Storage Manager tracks file system image backups on tape, but not individual files.

Tivoli Data Protection for NDMP and Other NAS Backup Methods

Using Tivoli Data Protection for NDMP to protect NAS file servers allows Tivoli Storage Manager to control operations while the NAS file server transfers the data. To use a backup-archive client to back up a NAS file server, mount the NAS file server file system on the client machine (with either an NFS mount or a CIFS map) and back up as usual. The following table compares the two methods:

Table 3. Comparing Tivoli Data Protection for NDMP and Tivoli Storage Manager Backup-Archive Client

Tivoli Data Protection for NDMP	Tivoli Storage Manager Backup-Archive Client
Network data traffic is less because the Tivoli Storage Manager server controls operations remotely, but the NAS file server moves the data locally.	Network data traffic is greater because all backup data goes across the LAN from the NAS file server to the client and then to the Tivoli Storage Manager server.
Less file server processing is required to back up a file system because the backup does not use file access protocols such as NFS and CIFS.	More file server processing is required because file backups require additional overhead for file access protocols such as NFS and CIFS.
The Tivoli Storage Manager server can be distant from the NAS file server and the tape library.	The Tivoli Storage Manager server must be within SCSI or Fibre Channel range of the tape library.

Tivoli Data Protection for NDMP Backups

When Tivoli Data Protection for NDMP backs up a NAS file server, it creates NAS file system image backups. The image backups are different from traditional Tivoli Storage Manager backups because the NAS file server transfers the data to the drives in the library. NAS file system image backups can be either full or differential image backups. The first backup of a file system on a NAS file server is always a full image backup. By default, subsequent backups are differential image backups containing only data that has changed in the file system since the last full image backup. If a full image backup does not already exist, a full image backup is performed.

If you restore a differential image, Tivoli Storage Manager automatically restores the full backup image first, followed by the differential image. Tivoli Storage Manager cannot restore individual files from full or differential image backups.

The following operations are not supported for data that has been backed up by using Tivoli Data Protection for NDMP:

- Storage pool migration
- Storage pool backup and restore
- Reclamation
- Move data operations
- Export and import operations
- Backup set generation

Multiple Tivoli Storage Manager Servers Sharing Libraries

Figure 3 on page 36 shows a SAN configuration in which two Tivoli Storage Manager servers share a library.

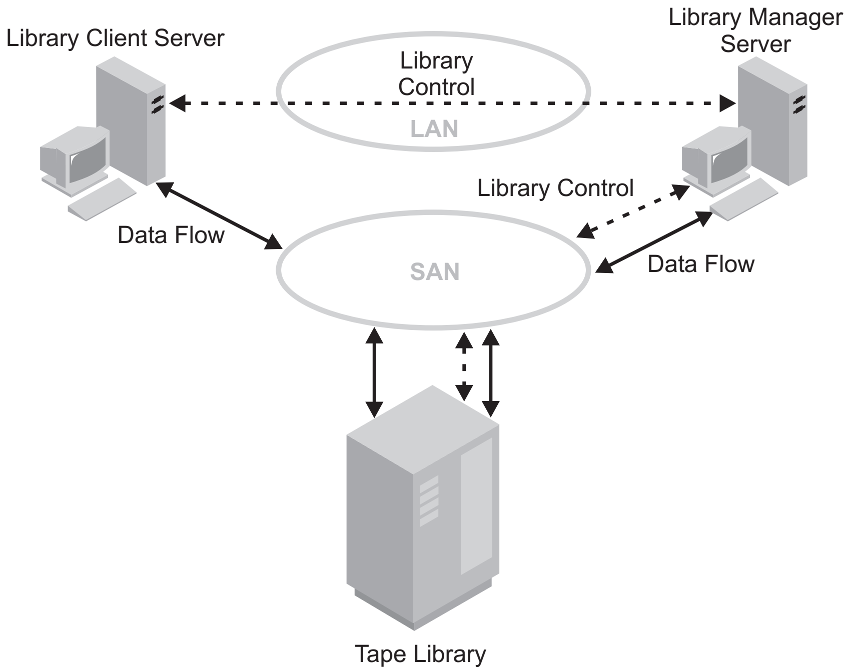


Figure 3. Library Sharing in a Storage Area Network (SAN) Configuration. The servers communicate over the LAN. The library manager controls the library over the SAN. The library client stores data to the library devices over the SAN.

When Tivoli Storage Manager servers share a library, one server, the *library manager*, controls device operations. These operations include mount, dismount, volume ownership, and library inventory. Other servers, *library clients*, use server-to-server communications to contact the library manager and request device service. Data moves over the SAN between each server and the storage device.

Tivoli Storage Manager servers use the following features when sharing an automated library:

Partitioning of the Volume Inventory
The inventory of media volumes in the shared library is partitioned among servers. Either one server owns a particular volume, or the volume is in the global scratch pool. No server owns the scratch pool at any given time.

Serialized Drive Access
Only one server accesses each tape drive at a time. Drive access is serialized and controlled so that servers do not dismount other servers' volumes or write to drives where other servers mount their volumes.

Serialized Mount Access
The library autochanger performs a single mount or dismount operation at a time. A single server (library manager) performs all mount operations to provide this serialization.

LAN-Free Data Movement

Tivoli Storage Manager allows a client, through a storage agent, to directly back up and restore data to a tape library on a SAN. Figure 4 on page 37 shows a SAN configuration in which a client directly accesses a tape, disk, or FILE library to read or write data.

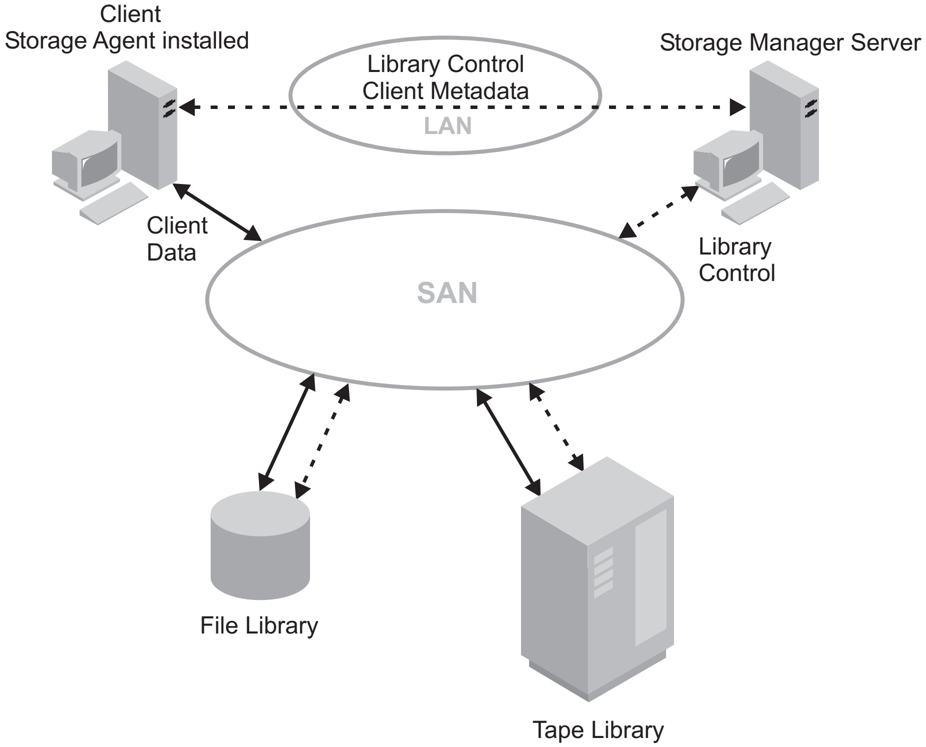


Figure 4. LAN-Free Data Movement. Client and server communicate over the LAN. The server controls the device on the SAN. Client data moves over the SAN to the device.

LAN-free data movement requires the installation of a storage agent on the client machine. The server maintains the database and recovery log, and acts as the library manager to control device operations. The storage agent on the client handles the data transfer to the device on the SAN. This implementation frees up bandwidth on the LAN that would otherwise be used for client data movement.

The following outlines a typical backup scenario for a client that uses LAN-free data movement:

1. The client begins a backup operation. The client and the server exchange policy information over the LAN to determine the destination of the backed up data.

   For a client using LAN-free data movement, the destination is a storage pool that uses a device on the SAN.
2. Because the destination is on the SAN, the client contacts the storage agent, which will handle the data transfer. The storage agent sends a request for a volume mount to the server.
3. The server contacts the storage device and, in the case of a tape library, mounts the appropriate media.
4. The server notifies the client of the location of the mounted media.
5. The client, through the storage agent, writes the backup data directly to the device over the SAN.

6. The storage agent sends file attribute information to the server, and the server stores the information in its database.

If a failure occurs on the SAN path, failover occurs. The client uses its LAN connection to the Tivoli Storage Manager server and moves the client data over the LAN.

Note: See the Tivoli Storage Manager home page at http://www.tivoli.com/support/storage_mgr/tivolimain.html for the latest information on clients that support the feature.

Planning for Server Storage

This section discusses how to evaluate your environment to determine the device classes and storage pools for your server storage.

1. Determine the storage devices that are available to the server. For example, determine how many tape drives you have that you will allow the server to use.

   The servers can share devices in libraries that are attached through a SAN. If the devices are not on a SAN, the server expects to have exclusive use of the drives defined to it. If another application (including another Tivoli Storage Manager server) tries to use a drive while the server to which the drive is defined is running, some server functions may fail.
2. Determine the device type and device class for each of the available devices. Group together similar devices and identify their device classes. For example, create separate categories for 4mm and 8mm devices.

   Note: For sequential access devices, you can categorize the type of removable media based on their capacity. For example, standard length cartridge tapes and longer length cartridge tapes require different device classes.
3. Determine how the mounting of volumes is accomplished for the devices:
   - Devices that require operators to load volumes must be part of a defined MANUAL library.
   - Devices that are automatically loaded must be part of a defined SCSI or 349X. Each automated library device is a separate library.
   - Devices that are controlled by StorageTek Automated Cartridge System Library Software (ACSLS) must be part of a defined ACSLS library.
   - Devices that are managed by an external media management system must be part of a defined EXTERNAL library.
4. If you are considering storing data for one Tivoli Storage Manager server using the storage of another Tivoli Storage Manager server, consider network bandwidth and network traffic. If your network resources constrain your environment, you may have problems using the SERVER device type efficiently.

   Also consider the storage resources available on the target server. Ensure that the target server has enough storage space and drives to handle the load from the source server.
5. Determine the storage pools to set up, based on the devices you have and on user requirements. Gather users' requirements for data availability. Determine which data needs quick access and which does not.
6. Be prepared to label removable media. You may want to create a new labeling convention for media so that you can distinguish them from media used for other purposes.

How Tivoli Storage Manager Uses and Reuses Removable Media

Tivoli Storage Manager allows you to control how removable media are used and reused. After Tivoli Storage Manager selects an available medium, that medium is used and eventually reclaimed according to its associated policy.

Tivoli Storage Manager manages the data on the media, but you manage the media itself, or you can use a removable media manager. Regardless of the method used, managing media involves creating a policy to expire data after a certain period of time or under certain conditions, move valid data onto new media, and reuse the empty media.

In addition to information about storage pool volumes, the volume history contains information about tapes used for database backups and exports (for disaster recovery purposes). The process for reusing these tapes is slightly different from the process for reusing tapes containing client data backups.

Figure 5 shows a typical life cycle for removable media. The numbers (such as **1**) refer to numbers in the figure.

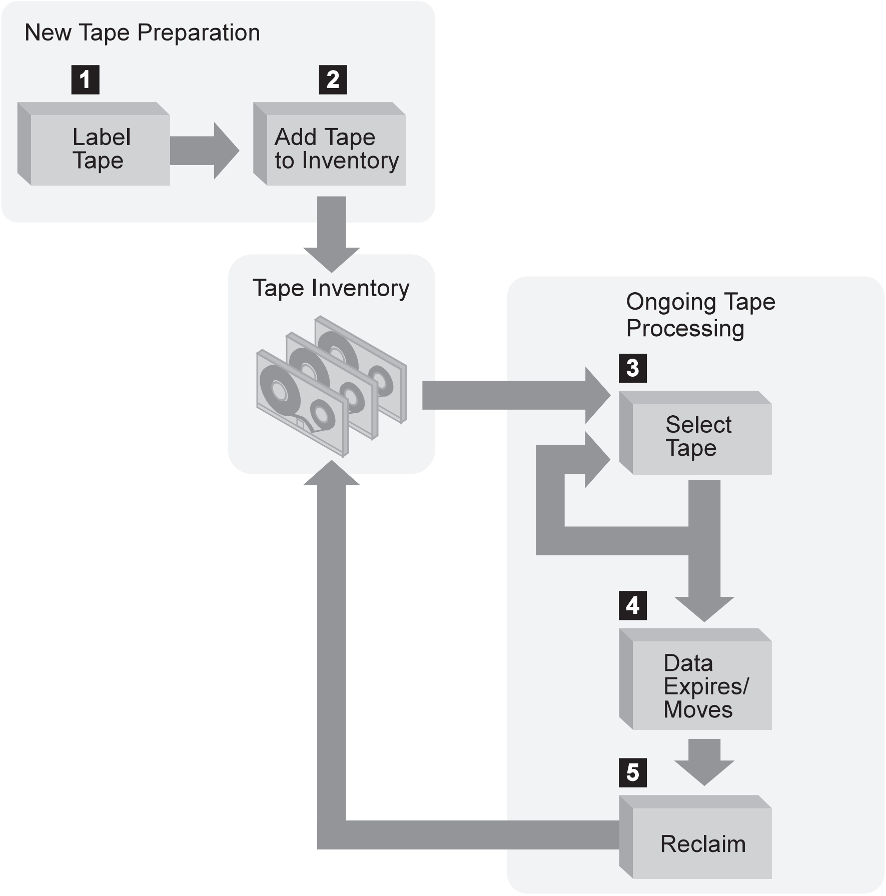


Figure 5. Simplified View of the Life Cycle of a Tape

1. You label **1** and check in **2** the media. Checking media into a manual library simply means storing them (for example, on shelves). Checking media into an automated library involves adding them to the library volume inventory.

See “Labeling Removable Media Volumes” on page 114.

2. If you plan to define volumes to a storage pool associated with a device, you should check in the volume with its status specified as private. Use of scratch volumes is more convenient in most cases.
3. A client sends data to the server for backup, archive, or space management. The server stores the client data on the volume. Which volume the server selects **3** depends on:
   - The policy domain to which the client is assigned.
   - The management class for the data (either the default management class for the policy set, or the class specified by the client in the client’s include/exclude list or file).
   - The storage pool specified as the destination in either the management class (for space-managed data) or copy group (for backup or archive data). The storage pool is associated with a device class, which determines which device and which type of media is used.
   - Whether the maximum number of scratch volumes that a server can request from the storage pool has been reached when the scratch volumes are selected.
   - Whether collocation is enabled for that storage pool. When collocation is enabled, the server attempts to place data for different clients or client nodes on separate volumes. For more information, see “Keeping a Client’s Files Together: Collocation” on page 184.

   See Figure 6.

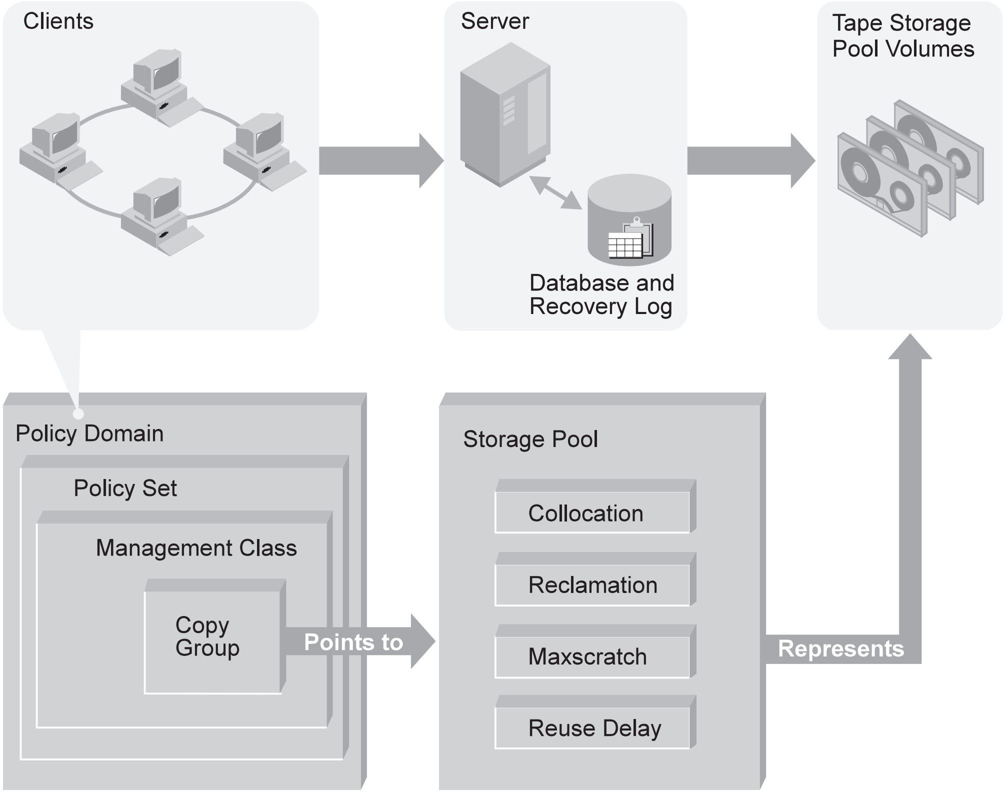


Figure 6. How Tivoli Storage Manager Affects Media Use

4. The data on a volume changes over time as a result of:

- Expiration of files **4** (affected by management class and copy group attributes, and the frequency of expiration processing). See “Basic Policy Planning” on page 270.
- Movement and deletion of file spaces by administrator
- Automatic reclamation of media **5**

  The amount of data on the volume and the reclamation threshold set for the storage pool affects when the volume is reclaimed. When the volume is reclaimed, any valid, unexpired data is moved to other volumes or possibly to another storage pool (for storage pools with single-drive libraries)..
- Collocation, by which Tivoli Storage Manager attempts to keep data belonging to a single client node or a single client file space on a minimal number of removable media in a storage pool.

If the volume becomes empty because all valid data either expires or is moved to another volume, the volume is available for reuse (unless a time delay has been specified for the storage pool). The empty volume becomes a scratch volume if it was initially a scratch volume. The volume starts again at step 3 on page 40.

5. You determine when the media has reached its end of life.

   For volumes that you defined (private volumes), check the statistics on the volumes by querying the database. The statistics include the number of write passes on a volume (compare with the number of write passes recommended by the manufacturer) and the number of errors on the volume.

   You must move any valid data off a volume that has reached end of life. Then, if the volume is in an automated library, check out the volume from the library. If the volume is not a scratch volume, delete the volume from the database.

Tivoli Storage Manager Volumes

Tivoli Storage Manager classifies its volumes into two categories: *private* and *scratch*.

A private volume is a labeled volume that is in use or owned by an application, and may contain valid data. You must define each private volume, and it can only be used to satisfy a request to mount that volume by name. Private volumes do not return to scratch when they become empty. For information on defining volumes, see “Defining Storage Pool Volumes” on page 167. For information on changing the status of a volume in an automated library, see “Changing the Status of a Volume” on page 124.

A scratch volume is a labeled volume that is empty or contains no valid data, and can be used to satisfy any request to mount a scratch volume. When data is written to a scratch volume, its status is changed to private, and it is defined as part of the storage pool for which the mount request was made. When valid data is moved from the volume and the volume is reclaimed, the volume returns to scratch status and can be reused by any storage pool associated with the library.

For each storage pool, you must decide whether to use scratch volumes. A scratch volume is selected for a mount request only if scratch volumes are allowed in the storage pool. If you do not use scratch volumes, you must define each volume. Tivoli Storage Manager keeps an inventory of volumes in each automated library it manages and tracks whether the volumes are in scratch or private status. If a storage pool contains scratch volumes, the server can choose a scratch volume from those that have been checked into the library.

Any storage pools associated with the same automated library can dynamically acquire volumes from the library's pool of scratch volumes. You do not need to allocate volumes to the different storage pools. Even if only one storage pool is associated with a library, you do not need to explicitly define all the volumes for the storage pool. Volumes are automatically added to and deleted from the storage pool by the server.

Note: A disadvantage of using scratch volumes is that volume usage information, which you can use to determine when the media has reached its end of life, is deleted when the private volume is returned to the scratch volume pool.

The Volume Inventory for an Automated Library

A library's volume inventory includes only those volumes that have been checked into that library. This inventory is not necessarily identical to the list of volumes in the storage pools associated with the library. For example:

- A volume can be checked into the library but not be in a storage pool (a scratch volume, a database backup volume, or a backup set volume).
- A volume can be defined to a storage pool associated with the library (a private volume), but not checked into the library.

For more information on how to check in volumes, see "Checking New Volumes into a Library" on page 117.

Tivoli Storage Manager Storage Objects

The Tivoli Storage Manager devices and media are represented by objects that you define and that are stored in the database. The objects contain information about the devices and media. You can query, update, and delete the objects. The storage objects are:

- Library
- Drive
- Device class
- Data mover
- Path
- Server
- Storage pool
- Storage pool volume

The following sections describe these objects.

Library, Drive, and Device Class

These three objects taken together represent a physical storage entity as shown in Figure 7 on page 43.

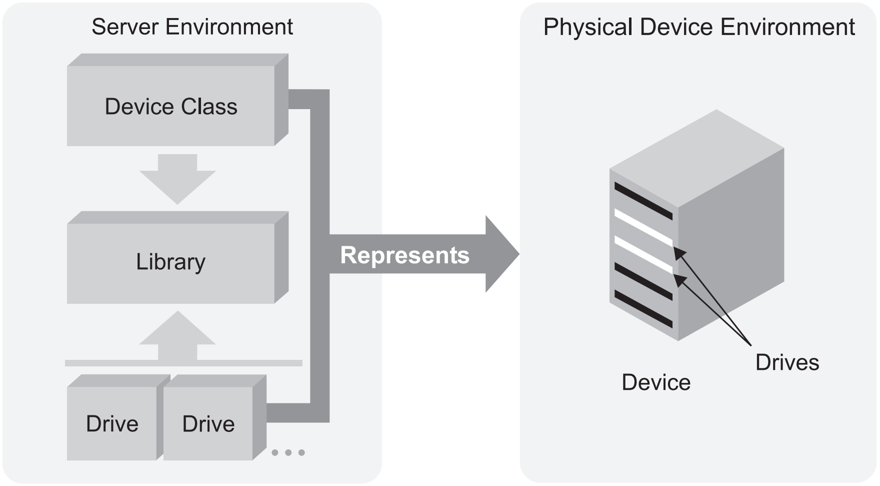


Figure 7. Removable Media Devices Are Represented by a Library, Drive, and Device Class

A physical library is a collection of one or more drives that share similar media mounting requirements. That is, the drive may be mounted by an operator or by an automated mounting mechanism. A library object definition specifies the library type (for example, SCSI or 349X) and other characteristics associated with the library type (for example, the category numbers used by an IBM 3494 library for private and scratch volumes).

Each drive mechanism within a device that uses removable media is represented by a drive object. For devices with multiple drives, including automated libraries, each drive is separately defined and must be associated with a library. Drive definitions can include such information as the element address (for drives in SCSI libraries), how often the drive is cleaned (for tape drives), and whether or not the drive is online.

Each device defined to Tivoli Storage Manager is associated with one device class. That device class specifies a device type and media management information, such as recording format, estimated capacity, and labeling prefixes. Device classes for a tape or optical drives, must also specify a library.

A device type identifies a device as a member of a group of devices that share similar media characteristics. For example, the 8MM device type applies to 8mm tape drives. Device types include a variety of removable media types and also FILE and SERVER:

Removable Media
Tivoli Storage Manager provides a set of specified removable media device types, such as 8MM for 8mm tape devices, or REMOVABLEFILE for Jaz or Zip drives. The GENERICTAPE device type is provided to support certain devices that do not use the Tivoli Storage Manager device driver. See "Chapter 8. Defining Device Classes" on page 141 and *Administrator's Reference* for more information about supported removable media device types.

FILE This device type allows you to create sequential volumes by creating files on disk storage. To the server, these files have the characteristics of a tape volume. The FILE device type does not require you to define library or drive objects; only a device class is required.

You can use FILE volumes as a way to use disk storage without having to define volumes. FILE volumes can also be useful when transferring data for purposes such as electronic vaulting.

SERVER

This device type allows you to create volumes for one Tivoli Storage Manager server that exist as archived files in the storage hierarchy of another server. These virtual volumes have the characteristics of sequential access volumes such as tape. You must define a device class and a target server. No library or drive definition is required.

Virtual volumes can be used for the following:

- Device-sharing between servers. One server is attached to a large tape library device. Other servers can use that library device indirectly through a SERVER device class.
- Data-sharing between servers. By using a SERVER device class to export and import data, physical media remains at the original location instead having to be transported.
- Immediate offsite storage. Storage pools and databases can be backed up without physically moving media to other locations.
- Offsite storage of the Tivoli Disaster Recovery Manager (DRM) recovery plan file.
- Electronic vaulting.

See "Using Virtual Volumes to Store Data on Another Server" on page 469.

- For more information about the drive object, see "Defining Drives" on page 94 and "Managing Drives" on page 133.
- For more information about the library object, see "Defining Libraries" on page 93 and "Managing Libraries" on page 131.
- For more information about the device class object, see "Chapter 8. Defining Device Classes" on page 141.

Storage Pool and Storage Pool Volume

A storage pool is a collection of storage pool volumes that are associated with one device class and one media type. For example, a storage pool that is associated with a device class for 8mm tape volumes contains only 8mm tape volumes. You can control the characteristics of storage pools, such as whether scratch volumes are used. For details about defining storage pools, see "Chapter 9. Managing Storage Pools and Volumes" on page 155.

Tivoli Storage Manager supplies default disk storage pools. For more information, see "Configuring Random Access Volumes on Disk Devices" on page 52.

Figure 8 on page 45 shows storage pool volumes grouped into a storage pool. Each storage pool represents only one type of media. For example, a storage pool for 8mm devices represents collections of only 8mm tapes.

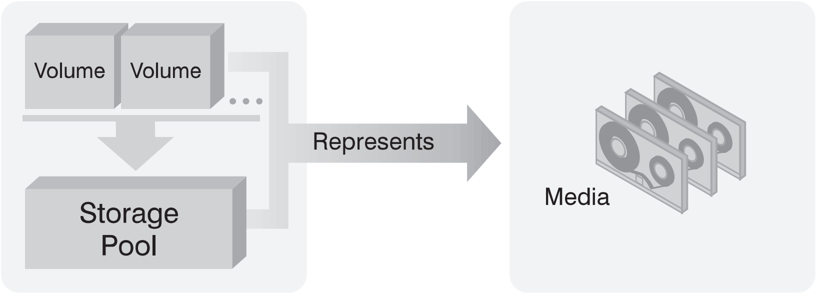


Figure 8. Relationships of Storage Pool Volumes, Storage Pools, and Media

For DISK device classes, you must define volumes. For other device classes, such as tape and FILE, you can allow the server to dynamically acquire scratch volumes and define those volumes as needed. For details, see “Preparing Volumes for Random Access Storage Pools” on page 166 and “Preparing Volumes for Sequential Access Storage Pools” on page 166.

One or more device classes are associated with one *library*, which can contain multiple drives. When you define a storage pool, you associate the pool with a device class. Volumes are associated with pools. Figure 9 shows these relationships.

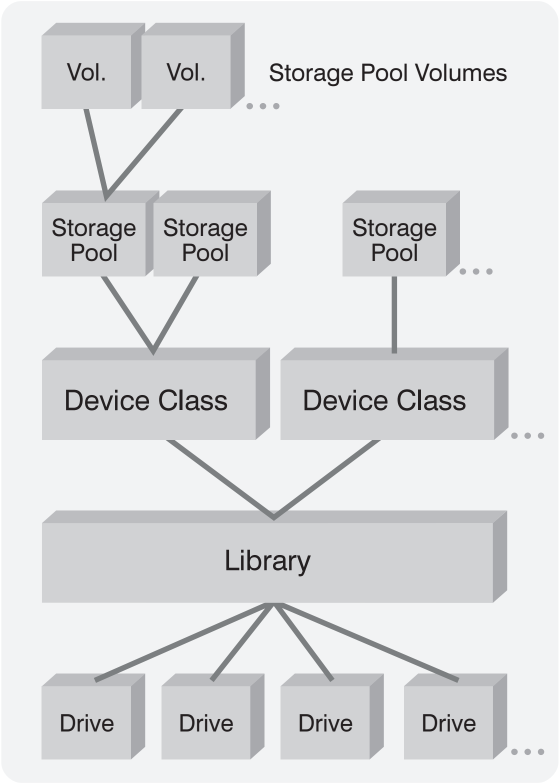


Figure 9. Relationships between Storage and Device Objects

For more information about the storage pool and volume objects, see “Chapter 9. Managing Storage Pools and Volumes” on page 155.

Data Mover

Data movers are devices that accept requests to transfer data from applications such as Tivoli Storage Manager. Data movers transfer data:

- Between a source storage device to a target storage device
- Without using significant Tivoli Storage Manager server or client resources
- Without using significant LAN network resources

You must define a data mover if you will be using network-attached storage. Among other characteristics, you must specify the data format that will be used and the data mover address.

For more information about the data mover object, see “Defining Data Movers” on page 94 and “Managing Data Movers” on page 139.

Path

Paths allow access to drives and libraries and drives. A path definition specifies a source and a destination. The source accesses the destination, but data can flow in either direction between the source and destination. Here are a few examples of paths:

- In LAN operations and most SAN operations, control data and client data flows across a path from the Tivoli Storage Manager server to an automated library that is defined to the server. During a restore, a client data also flows from the library back to the server.
- In NDMP operations backup data flows across a path between the source, a data mover defined for a NAS file server, and the destination, a tape drive. Restore data flows back across the path from the tape drive to the NAS file server. Paths can address the same destination device from different sources. For example, in NDMP operations, paths are always required between the data movers that represent NAS file servers and the drives to which the file servers transfer data for backup. Paths can point from multiple NAS file server data movers to the same tape drive.

For more information about the path object, see “Defining Paths” on page 95 and “Managing Paths” on page 138.

Server

You need to define a server object, for the following purposes:

- To use a library that is on a SAN and that is managed by another Tivoli Storage Manager server. You must define that server and then specify it as the library manager when you define the library. For more information, “Setting up the Library Client Servers” on page 73.
- To use LAN-free data movement. You define the storage agent as a server. For more information, see *Tivoli Storage Manager Managed System for SAN Storage Agent User’s Guide*.
- To store client data in the storage of another Tivoli Storage Manager server. For more information, see “Using Virtual Volumes to Store Data on Another Server” on page 469.

Among other characteristics, you must specify the server TCP/IP address.

How Tivoli Storage Manager Mounts and Dismounts Removable Media

When data is to be stored in or retrieved from a storage pool, the server does the following:

1. Selects a volume from the storage pool. The selection is based on the type of operation:

   Retrieval
   The name of the volume that contains the data to be retrieved is stored in the database.

   Store If a defined volume in the storage pool can be used, the server selects that volume.

   If no defined volumes in the storage pool can be used, and if the storage pool allows it, the server selects a scratch volume.
2. Checks the device class associated with the storage pool to determine the name of the library that contains the drives to be used for the operation.
   - The server searches the library for an available drive or until all drives have been checked. A drive status can be:
     - Offline.
     - Busy and not available for the mount.
     - In an error state and not available for the mount.
     - Online and available for the mount.
3. Mounts the volume:
   - For a manual library, the server displays a mount message for a private or a scratch volume to be mounted in the selected drive.
   - For an automated library, the server directs the library to move the volume from a storage slot into the selected drive. No manual intervention is required.

     If a scratch mount is requested, the server checks the library’s volume inventory for a scratch volume. If one is found, its status is changed to private, it is mounted in the drive, and it is automatically defined as part of the original storage pool. However, if the library’s volume inventory does not contain any scratch volumes, the mount request fails.
4. Dismounts the volume when it has finished accessing the volume and the mount retention period has elapsed.
   - For a manual library, the server ejects the volume from the drive so that an operator can place it in its storage location.
   - For an automated library, the server directs the library to move the volume from the drive back to its original storage slot in the library.

Configuring Devices

Before the Tivoli Storage Manager server can used a device, the device must be configured to the operating system as well as to the server. Table 4 summarizes the definitions that are required for different device types.

Table 4. Required Definitions for Storage Devices

		Required Definitions		
Device	**Device Types**	**Library**	**Drive**	**Device Class**
Magnetic disk	DISK	—	—	Yes [1]
	FILE	—	—	Yes
Tape	3570 3590 4MM 8MM CARTRIDGE [2,3] DLT DTF ECARTRIDGE [3] GENERICTAPE LTO QIC	Yes	Yes	Yes
Optical	OPTICAL WORM WORM12 WORM14	Yes	Yes	Yes
Removable media (file system)	REMOVABLEMEDIA	Yes	Yes	Yes
Virtual volumes	SERVER	—	—	Yes

[1] The DISK device class exists at installation and cannot be changed.

[2] The CARTRIDGE device type is for IBM 3480, 3490, and 3490E tape drives.

[3] The ECARTRIDGE device type is for cartridge tape drives such as the StorageTek SD-3, 9480, 9890, and 9940 drives.

Mapping Devices to Device Classes

As an example of mapping devices to device classes, assume that you have the following devices to use for server storage:

- Internal disk drives
- An automated tape library with 8mm drives
- A manual DLT tape drive

You can map storage devices to device classes as shown in Table 5.

Table 5. Mapping Storage Devices to Device Classes

Device Class	Description
DISK	Storage volumes that reside on the internal disk drive Tivoli Storage Manager provides one DISK device class that is already defined. You do not need and cannot define another device class for disk storage.
8MM_CLASS	Storage volumes that are 8mm tapes, used with the drives in the automated library
DLT_CLASS	Storage volumes that are DLT tapes, used on the DLT drive

You must define any device classes that you need for your removable media devices such as tape drives. See “Chapter 8. Defining Device Classes” on page 141 for information on defining device classes to support your physical storage environment.

Mapping Storage Pools to Device Classes and Devices

After you have categorized your storage devices, identify availability, space, and performance requirements for client data that is stored in server storage. These requirements help you determine where to store data for different groups of clients and different types of data. You can then create storage pools that are storage destinations for backed-up, archived, or space-managed files to match requirements.

For example, you determine that users in the business department have three requirements:

- Immediate access to certain backed-up files, such as accounts receivable and payroll accounts.

  These files should be stored on disk. However, you need to ensure that data is moved from the disk to prevent it from becoming full. You can set up a storage hierarchy so that files can migrate automatically from disk to the automated tape library.
- Periodic access to some archived files, such as monthly sales and inventory reports.

  These files can be stored on 8mm tapes, using the automated library.
- Occasional access to backed-up or archived files that are rarely modified, such as yearly revenue reports.

  These files can be stored using the DLT drive.

To match user requirements to storage devices, you define storage pools, device classes, and, for device types that require them, libraries and drives. For example, to set up the storage hierarchy so that data migrates from the BACKUPPOOL to 8mm tapes, you specify BACKTAPE1 as the next storage pool for BACKUPPOOL. See Table 6.

Table 6. Mapping Storage Pools to Device Classes, Libraries, and Drives

Storage Pool	Device Class	Library (Hardware)	Drives	Volume Type	Storage Destination
BACKUPPOOL	DISK	—	—	Storage volumes on the internal disk drive	For a backup copy group for files requiring immediate access
BACKTAPE1	8MM_CLASS	AUTO_8MM (Exabyte EXB-210)	DRIVE01, DRIVE02	8mm tapes	For overflow from the BACKUPPOOL and for archived data that is periodically accessed
BACKTAPE2	DLT_CLASS	MANUAL_LIB (Manually mounted)	DRIVE03	DLT tapes	For backup copy groups for files that are occasionally accessed

Note: Tivoli Storage Manager has default disk storage pools named BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL. For more information, see “Configuring Random Access Volumes on Disk Devices” on page 52.

Chapter 3. Using Magnetic Disk Devices

Tivoli Storage Manager uses magnetic disk devices to do the following:

- Store the database and the recovery log. For details, see “Chapter 18. Managing the Database and Recovery Log” on page 383.
- Store client data that has been backed up, archived, or migrated from client nodes. The client data is stored in storage pools. Procedures for configuring disk storage of client data are described in this chapter.
- Store backups of the database and export and import data. See “Using FILE Volumes for Database Backups and Export Operations” on page 55.

See the following sections:

Tasks:
“Configuring Random Access Volumes on Disk Devices” on page 52
“Configuring FILE Sequential Volumes on Disk Devices” on page 53
“Varying Disk Volumes Online or Offline” on page 53
“Using Cache” on page 54
“Freeing Space on Disk” on page 54
“Specifying Scratch FILE Volumes” on page 54
“Using FILE Volumes for Database Backups and Export Operations” on page 55

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator’s Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Note: Some of the tasks described in this chapter require an understanding of Tivoli Storage Manager storage objects. For an introduction to these storage objects, see “Tivoli Storage Manager Storage Objects” on page 42.

Configuring Disk Devices

Tivoli Storage Manager stores data on magnetic disks in two ways:

- In random access volumes, as data is normally stored on disk. See “Configuring Random Access Volumes on Disk Devices” on page 52.
- In files on the disk that are treated as sequential access volumes. See “Configuring FILE Sequential Volumes on Disk Devices” on page 53.

© Copyright IBM Corp. 1993, 2002

Task	Required Privilege Class
Configuring Random Access Volumes on Disk Devices Configuring FILE Sequential Volumes on Disk Devices	System

Configuring Random Access Volumes on Disk Devices

Tivoli Storage Manager provides a defined DISK device class that is used with all disk devices.

Tip: For performance reasons, define storage pool volumes on disk drives that reside on the server machine, not on remotely mounted file systems.

Do the following to use random access volumes on a disk device:

1. Define a storage pool that is associated with the DISK device class, or use one of the default storage pools that Tivoli Storage Manager provides (ARCHIVEPOOL, BACKUPPOOL, and SPACEMGPOOL).

   For example, enter the following command on the command line of an administrative client:

   ```
   define stgpool engback1 disk maxsize=5m highmig=85 lowmig=40
   ```

   This command defines storage pool ENGBACK1.

   See “Example: Defining Storage Pools” on page 161 for details.
2. Prepare a volume for use in a random access storage pool by defining the volume. For example, you want to define a 21MB volume for the ENGBACK1 storage pool. You want the volume to be located in the path */usr/tivoli/tsm/server/bin* and named stgvol.002. Enter the following command:

   ```
   define volume engback1 /usr/tivoli/tsm/server/bin/stgvol.002 formatsize=21
   ```

   If you do not specify a full path name, the command uses the current path. See “Defining Storage Pool Volumes” on page 167 for details.

   This one-step process replaces the former two-step process of first formatting a volume (using DSMFMT) and then defining the volume. If you choose to use the two-step process, the DSMFMT utility is available from the operating system command line. See *Administrator’s Reference* for details.

   Another option for preparing a volume is to create a raw logical volume by using SMIT.
3. Do one of the following:
   - Specify the new storage pool as the destination for client files that are backed up, archived, or migrated, by modifying existing policy or creating new policy. See “Chapter 12. Implementing Policies for Client Data” on page 269 for details.
   - Place the new storage pool in the storage pool migration hierarchy by updating an already defined storage pool. See “Example: Updating Storage Pools” on page 162.

Configuring FILE Sequential Volumes on Disk Devices

Another way to use magnetic disk storage is to use files as volumes that store data sequentially (as on tape volumes). You can use FILE sequential volumes to transfer data for purposes such as electronic vaulting. For example, you can send the results of an export operation or a database backup operation to another location. At the receiving site, the files can be placed on tape or disk. You can define a device class with a device type of FILE.

To use files as volumes that store data sequentially, do the following:

1. Define a device class with device type FILE.

   For example, enter the following command on the command line of an administrative client:

   ```
   define devclass fileclass devtype=file mountlimit=2
   ```

   This command defines device class FILECLASS with a device type of FILE.

   See “Defining and Updating FILE Device Classes” on page 148.

   To store database backups or exports on FILE volumes, this step is all you need to do to prepare the volumes. For more information, see “Defining Device Classes for Backups” on page 511 and “Planning for Sequential Media Used to Export Data” on page 479.
2. Define a storage pool that is associated with the new FILE device class.

   For example, enter the following command on the command line of an administrative client:

   ```
   define stgpool engback2 fileclass maxscratch=100 mountlimit=2
   ```

   This command defines storage pool ENGBACK2 with device class FILECLASS.

   See “Defining or Updating Primary Storage Pools” on page 158 for details.

   To allow Tivoli Storage Manager to use scratch volumes for this device class, specify a value greater than zero for the number of maximum scratch volumes when you define the device class. If you do set MAXSCRATCH=0 to not allow scratch volumes, you must define each volume to be used in this device class. See “Preparing Volumes for Sequential Access Storage Pools” on page 166 for details.
3. Do one of the following:
   - Specify the new storage pool as the destination for client files that are backed up, archived, or migrated, by modifying existing policy or creating new policy. See “Chapter 12. Implementing Policies for Client Data” on page 269 for details.
   - Place the new storage pool in the storage pool migration hierarchy by updating an already defined storage pool. See “Example: Updating Storage Pools” on page 162.

Varying Disk Volumes Online or Offline

Task	Required Privilege Class
Vary a disk volume online or offline	System or operator

To perform maintenance on a disk volume or to upgrade disk hardware, you can vary a disk volume offline. For example, to vary the disk volume named */storage/pool001* offline, enter:

```
vary offline /storage/pool001
```

If Tivoli Storage Manager encounters a problem with a disk volume, the server automatically varies the volume offline.

You can make the disk volume available to the server again by varying the volume online. For example, to make the disk volume named */storage/pool001* available to the server, enter:

```
vary online /storage/pool001
```

Using Cache

When you define a storage pool that uses disk random access volumes, you can choose to enable or disable cache. When you use cache, a copy of the file remains on disk storage even after the file has been migrated to the next pool in the storage hierarchy (for example, to tape). The file remains in cache until the space it occupies is needed to store new files.

Using cache can improve how fast a frequently accessed file is retrieved. Faster retrieval can be important for clients storing space-managed files. If the file needs to be accessed, the copy in cache can be used rather than the copy on tape. However, using cache can degrade the performance of client backup operations and increase the space needed for the database. For more information, see “Using Cache on Disk Storage Pools” on page 183.

Freeing Space on Disk

As client files expire, the space they occupy is not freed for other uses until you run expiration processing on the server.

Expiration processing deletes from the database information about any client files that are no longer valid according to the policies you have set. For example, suppose four backup versions of a file exist in server storage, and only three versions are allowed in the backup policy (the management class) for the file. Expiration processing deletes information about the oldest of the four versions of the file. The space that the file occupied in the storage pool becomes available for reuse.

You can run expiration processing by using one or both of the following methods:

- Use the EXPIRE INVENTORY command. See “Running Expiration Processing to Delete Expired Files” on page 302.
- Set the server option for the expiration interval, so that expiration processing runs periodically. See *Administrator’s Reference* for information on how to set the options.

Specifying Scratch FILE Volumes

You can specify a maximum number of scratch volumes for a storage pool that has a FILE device type. When the server needs a new volume, the server automatically creates a file that is a scratch volume, up to the number you specify. When scratch volumes used in storage pools become empty, the files are deleted.

Using FILE Volumes for Database Backups and Export Operations

When you back up the database or export server information, Tivoli Storage Manager records information about the volumes used for these operations in the *volume history*. Tivoli Storage Manager will not allow you to reuse these volumes until you delete the volume information from the volume history. To reuse volumes that have previously been used for database backup or export, use the DELETE VOLHISTORY command. For information about the volume history and volume history files, see "Saving the Volume History File" on page 514.

Note: If your server is licensed for the Tivoli Disaster Recovery Manager (DRM) product, the volume information is automatically deleted during MOVE DRMEDIA command processing. For additional information about DRM, see "Chapter 23. Using Tivoli Disaster Recovery Manager" on page 545.

Chapter 4. Attaching Devices to the Server System

For Tivoli Storage Manager to use a device, you must attach the device to your server system and install the appropriate device driver.

Tasks:
"Attaching a Manual Drive"
"Attaching an Automated Library Device" on page 58
"Installing and Configuring Device Drivers" on page 59

Devices Supported by Tivoli Storage Manager

A list of supported storage devices is available on the Tivoli Storage Manager Web site at the following address:

http://www.tivoli.com/support/storage_mgr/tivolimain.html

Tivoli Storage Manager supports a wide range of manual devices and automated library devices:

Tape Devices
Tivoli Storage Manager supports the manual and automated tape devices listed at the Tivoli Storage Manager Web site.

Disk Devices
Tivoli Storage Manager supports the disk devices listed at the Tivoli Storage Manager Web site.

Optical Disk Devices
Tivoli Storage Manager supports the manual and automated optical disk devices listed at the Tivoli Storage Manager Web site.

Other Removable File Devices
Tivoli Storage Manager supports the removable media devices (such as optical drives, CD-ROM devices, and Iomega Zip and Jaz drives) listed at the Tivoli Storage Manager Web site.

Storage Area Network (SAN) Devices
Tivoli Storage Manager supports devices in a storage area network (SAN) environment, but the devices must be supported by the Tivoli Storage Manager device driver. See the Tivoli Storage Manager Web site for information about supported Fibre Channel hardware and configurations.

Attaching a Manual Drive

1. Install the SCSI adapter card in your system, if not already installed.
2. Determine the SCSI IDs available on the SCSI adapter card to which you are attaching the device. Find one unused SCSI ID for each drive.
3. Follow the manufacturer's instructions to set the SCSI ID for the drive to the unused SCSI IDs that you found. You may have to set switches on the back of the device or set the IDs on the operator's panel.

© Copyright IBM Corp. 1993, 2002

Note: Each device that is connected in a chain to a single SCSI bus must be set to a unique SCSI ID. If each device does not have a unique SCSI ID, serious system problems can arise.

4. Follow the manufacturer's instructions to attach the device to your server system hardware.

   Attention:

   a. Power off your system before attaching a device to prevent damage to the hardware.
   b. Attach a terminator to the last device in the chain of devices connected on one SCSI adapter card.
5. Install the appropriate device drivers. See "Installing and Configuring Device Drivers" on page 59.
6. Find the device worksheet that applies to your device. See http://www.tivoli.com/support/storage_mgr/tivolimain.html.
7. Determine the name for the device and record the name on the device worksheet. The device name for a tape drive is a special file name.

   See "Determining Device Special File Names" on page 60 for details.

Keep the Worksheets: The information you record on the worksheets can help you when you need to perform operations such as adding volumes. Keep the worksheets for future reference.

Attaching an Automated Library Device

1. Install the SCSI adapter card in your system, if not already installed.
2. Determine the SCSI IDs available on the SCSI adapter card to which you are attaching the device. Find one unused SCSI ID for each drive, and one unused SCSI ID for the library or autochanger controller.

   Note: In some automated libraries, the drives and the autochanger share a single SCSI ID, but have different LUNs. For these libraries, only a single SCSI ID is required. Check the documentation for your device.
3. Follow the manufacturer's instructions to set the SCSI ID for the drives to the unused SCSI IDs that you found. You may have to set switches on the back of the device or set the IDs on the operator's panel.

   Note: Each device that is connected in a chain to a single SCSI bus must be set to a unique SCSI ID. If each device does not have a unique SCSI ID, serious system problems can arise.
4. Follow the manufacturer's instructions to attach the device to your server system hardware.

   Attention:

   a. Power off your system before attaching a device to prevent damage to the hardware.
   b. Attach a terminator to the last device in the chain of devices connected on one SCSI adapter card. Detailed instructions should be in the documentation that came with your hardware.
5. Install the appropriate device drivers. See "Installing and Configuring Device Drivers" on page 59.
6. Find the device worksheet that applies to your device. See http://www.tivoli.com/support/storage_mgr/tivolimain.html.

7. Determine the name for each drive and for the library, and record the names on the device worksheet.

   The device name for a tape drive is a special file name. See “Determining Device Special File Names” on page 60 for details.

Keep the Worksheets: The information you record on the worksheets can help you when you need to perform operations such as adding volumes to an autochanger. Keep the work sheets for future reference.

Setting the Library Mode

For the Tivoli Storage Manager server to access a SCSI library, set the device for the appropriate mode. This is usually called *random* mode; however, terminology may vary from one device to another. Refer to the documentation for your device to determine how to set it to the appropriate mode.

Notes:

1. Some libraries have front panel menus and displays that can be used for explicit operator requests. However, if you set the device to respond to such requests, it typically will not respond to Tivoli Storage Manager requests.
2. Some libraries can be placed in *sequential* mode, in which volumes are automatically mounted in drives by using a sequential approach. This mode conflicts with how Tivoli Storage Manager accesses the device.

Installing and Configuring Device Drivers

To use a device, you must install the appropriate device driver. Tivoli Storage Manager provides its own device driver for non-IBM devices. The IBM device driver Atape is supported for IBM devices. These device drivers are available on the ftp site ftp://ftp.software.ibm.com/storage/devdrvr/. Tivoli Storage Manager also supports third party vendor device drivers if the devices are associated with the GENERICTAPE device class and the hardware vendor also supports that device driver. Using a device class other than GENERICTAPE with a third party vendor device driver is not recommended.

IBM tape drives and libraries

Install the device driver that IBM supplies. See “Installing Device Drivers for IBM SCSI Tape Devices” on page 61, “Installing Device Drivers for IBM 349X Libraries” on page 62.

Non-IBM tape drives and autochangers

You must ensure that you have installed the appropriate device drivers. When you install Tivoli Storage Manager, you must choose whether to install the Tivoli Storage Manager device driver or the native operating system device driver for tape devices.

Optical devices

Install the TSM device drivers. See http://www.tivoli.com/support/storage_mgr/tivolimain.html and “Configuring Device Drivers for Non-IBM Tape or Optical Drives” on page 62.

Selecting Device Drivers

Table 7 on page 60 and Table 8 on page 60 list the device drivers needed for Tivoli Storage Manager drives and libraries.

Table 7. Device Drivers for Tape and Optical Drives

Device	Device Type	Library	Device Driver
4MM drive	4MM	External, Manual, SCSI	Tivoli Storage Manager device driver
8MM drive	8MM	External, Manual, SCSI	
DLT drive	DLT	External, Manual, SCSI	
DTF drive	DTF	External, Manual, SCSI	
QIC drive	QIC	External, Manual, SCSI	
STK SD3, 9490, 9840, 9940 drive	ECARTRIDGE	External, Manual, SCSI, ACSLS	
Optical drive	OPTICAL	External, Manual, SCSI	
WORM drive	WORM	External, Manual, SCSI	
IBM 3570 drive	3570	External, Manual, SCSI	Atape
IBM 3480, 3490, 3490E drive	CARTRIDGE	External, Manual, SCSI, ACSLS, 349X	
IBM 3590, 3590E drive	3590	External, Manual, SCSI, ACSLS, 349X	
IBM LTO 3580 drive	LTO	External, Manual, SCSI	

Table 8. Device Drivers for Automated Libraries

Device	Library Type	Device Driver
IBM MP 3570, 3575 Library	SCSI	Atape
IBM LTO 3581, 3583, 3584 Library	SCSI	
IBM 3494, 3495 Library	349X	atldd
Non-IBM SCSI libraries	SCSI	Tivoli Storage Manager device driver (tsmscsi)

Note: See http://www.tivoli.com/support/storage_mgr/tivolimain.html for a list of supported drives and libraries.

Determining Device Special File Names

To work with removable media devices, Tivoli Storage Manager needs the device's special file name. You specify the device special file name when you issue the DEFINE PATH commands for drives and libraries. You can use SMIT to get the device special file.

When a device configures successfully, a logical file name is returned. Table 9 on page 61 specifies the name of the device (or the special file name) that corresponds to the drive or library. In the examples, *x* denotes any number from 0 to 9.

Table 9. Device Examples

Device	**Device Example**	**Logical File Name**
Tape drives that are supported by the Tivoli Storage Manager device driver	/dev/mt*x*	mt*x*
SCSI-attached libraries that are supported by the Tivoli Storage Manager device driver	/dev/lb*x*	lb*x*
Optical drives that are supported by the Tivoli Storage Manager device driver	/dev/rop*x*	op*x*
Drives associated with the GENERICTAPE device type	/dev/rmt*x*	rmt*x*
IBM 3570 devices and the Automatic Cartridge Facility feature of the IBM 3590 B11 as a library	/dev/rmt*x*.smc	rmt*x*
IBM 3575, 3581, 3583, 3584 libraries	/dev/smc*x*	smc*x*
IBM 349X libraries	/dev/lmcp*x*	lmcp*x*
Mount point to use on REMOVABLEFILE device type (CD-ROM)	/dev/cd*x*	cd*x*
File system to use on REMOVABLEFILE device type (zip drive)	/zip	Not applicable

Installing Device Drivers for IBM SCSI Tape Devices

To install the device drivers for IBM 3490, 3570, 358X, and 3590 devices, see

- *IBM SCSI Tape Drive, Medium Changer, and Library Device Drivers: Installation and User's Guide* (GC35-0154)
- *IBM Ultrium Device Drivers: Installation and User's Guide* (GC32-0430)

After completing the procedure in the manual, you receive a message from the system:

- If you are installing the device driver for an IBM 3480 or 3490 tape device, you receive a message (logical filename) of the form:

  ```
  rmtx Available
  ```

  Use the value of *x*, which is assigned automatically by the system, to complete the Device Name field on the worksheet that applies to your device (see http://www.tivoli.com/support/storage_mgr/tivolimain.html). For example, if the message is *rmt0 Available*, the special file name for the device is */dev/rmt0*. Enter */dev/rmt0* in the Device Name field for the drive on the worksheet. Always use the */dev/* prefix with the name provided by the system.
- If you are installing the device driver for an IBM 3570, 3575, 3581, 3583, 3584, or 3590 Model B11, you receive a message of the form:

  ```
  rmtx Available
  ```

  or

  ```
  smcx Available
  ```

  Note the value of *x*, which is assigned automatically by the system.
  - The special file name for the drive is /dev/rmt*x*.
  - The special file name for the media changer device is */dev/smcx*. The filetype *smc* stands for SCSI media changer.

For example, if the message is *rmt0 Available*, enter */dev/rmt0* in the Device Name field for the drive. Enter */dev/smc0* in the Device Name field on the worksheet for the library's robotics. Always use the */dev/* prefix with the name provided by the system.

Note: For multidrive devices (for example, IBM 3570 Model B12 or B22, or IBM 3575), you need only one *smcx*. Although you will receive a */dev/smcx* for each *rmt* device in the library, you need only one *smc* for the library on the worksheet.

Installing Device Drivers for IBM 349X Libraries

For an IBM 3494 or 3495 Tape Library Dataserver, refer to *IBM SCSI Tape Drive, Medium Changer, and Library Device Drivers: Installation and User's Guide* (GC35-0154). After completing the procedure in the manual, you will receive a message (logical filename) of the form:

```
lmcpx Available
```

where *x* is a number assigned automatically by the system. Use this information to complete the Device Name field on your worksheet. For example, if the message is **lmcp0 Available**, enter */dev/lmcp0* on the worksheet in the Device Name field for the library. Always use the */dev/* prefix with the name provided by the system.

Configuring Device Drivers for Non-IBM Libraries

Use the procedure in this section to configure Tivoli Storage Manager device drivers for autochangers for non-IBM libraries.

Run the SMIT program to configure the device driver for each autochanger or robot:

1. Select **Devices**.
2. Select **TSM Devices**.
3. Select **Library/MediumChanger**.
4. Select **Add a Library/MediumChanger**.
5. Select the TSM-SCSI-LB for any TSM supported library.
6. Select the parent adapter to which you are connecting the device. This number is listed in the form: 00-0X, where X is the slot number location of the SCSI adapter card.
7. When prompted, enter the CONNECTION address of the device you are installing. The connection address is a two-digit number. The first digit is the SCSI ID (the value you recorded on the worksheet). The second digit is the device's SCSI logical unit number (LUN), which is usually zero, unless otherwise noted. The SCSI ID and LUN must be separated by a comma (,). For example, a connection address of 4,0 has a SCSI ID=4 and a LUN=0.
8. Click on the **DO** button.

   You will receive a message (logical filename) of the form **lbX Available**. Note the value of X, which is a number assigned automatically by the system. Use this information to complete the Device Name field on your worksheet.

   For example, if the message is **lb0 Available**, the Device Name field is */dev/lb0* on the worksheet. Always use the */dev/* prefix with the name provided by SMIT.

Configuring Device Drivers for Non-IBM Tape or Optical Drives

Use the procedure in this section to configure Tivoli Storage Manager device drivers for non-IBM tape or optical drives.

Attention: Tivoli Storage Manager cannot write over *tar* or *dd* tapes, but *tar* or *dd* can write over TSM tapes.

Note: Tape drives can be shared only when the drive is not defined or the server is not started. The MKSYSB command will not work if both Tivoli Storage Manager and AIX are sharing the same drive or drives. To use the operating system's native tape device driver in conjunction with a SCSI drive, the device must be configured to AIX first and then configured to Tivoli Storage Manager. See your AIX documentation regarding these native device drivers.

Run the SMIT program to configure the device driver for each drive (including drives in libraries) as follows:

1. Select **Devices**.
2. Select **TSM Devices**.
3. Select **Tape Drive** or **Optical R/W Disk Drive**, depending on whether the drive is tape or optical.
4. Select **Add a Tape Drive** or **Add an Optical Disk Drive**, depending on whether the drive is tape or optical.
5. Select the TSM-SCSI-MT for any supported tape drive or TSM-SCSI-OP for any supported optical drive.
6. Select the adapter to which you are connecting the device. This number is listed in the form: 00-0X, where X is the slot number location of the SCSI adapter card.
7. When prompted, enter the CONNECTION address of the device you are installing. The connection address is a two-digit number. The first digit is the SCSI ID (the value you recorded on the worksheet). The second digit is the device's SCSI logical unit number (LUN), which is usually zero, unless otherwise noted. The SCSI ID and LUN must be separated by a comma (,). For example, a connection address of 4,0 has a SCSI ID=4 and a LUN=0.
8. Click on the **DO** button. You will receive a message:
   - If you are configuring the device driver for a tape device (other than an IBM tape drive), you will receive a message (logical filename) of the form **mtX Available**. Note the value of X, which is a number assigned automatically by the system. Use this information to complete the Device Name field on the worksheet.

     For example, if the message is **mt0 Available**, the Device Name field is */dev/mt0* on the worksheet. Always use the */dev/* prefix with the name provided by SMIT.
   - If you are configuring the device driver for an optical device, you will receive a message of the form **opX Available**. Note the value of X, which is a number assigned automatically by the system. Use this information to complete the Device Name field on the worksheet.

     For example, if the message is **op0 Available**, the Device Name field is */dev/rop0* on the worksheet. Always use the */dev/r* prefix with the name provided by SMIT.

Managing SCSI Devices and Fibre Channel Devices

The Tivoli Storage Manager device definition menus and prompts in SMIT allow for the management of both SCSI and FCP attached devices. The main menu for Tivoli Storage Manager has two options:

SCSI Attached Devices

Use this option to configure SCSI devices that are connected to a SCSI adapter in the host. The subsequent menus which have not changed configure devices

Fibre Channel system area network (SAN) Attached Devices

Use this option to configure devices that are connected to a Fibre Channel adapter in the host. Choose one of the following:

List Attributes of a Discovered Device

Lists attributes of a device known to the current ODM database.

FC Port ID:

This is the 24-bit FC Port ID(N(L)_Port or F(L)_Port). This is the address identifier that is unique within the associated topology where the device is connected. In the switch or fabric environments, it is usually determined by the switch, with the upper 2-bytes which are non-zero. In a Private Arbitrated Loop, it is the Arbitrated Loop Physical Address(AL_PA), with the upper 2-bytes being zero. Please consult with your FC vendors to find out how an AL_PA or a Port ID is assigned

Mapped LUN ID

This is from an FC to SCSI bridge (also, called a converter, router, or gateway) box, such as an IBM SAN Data Gateway (SDG) or Crossroads 4200. Please consult with your bridge vendors about how LUNs are mapped. It is recommended that you do not change LUN Mapped IDs.

WW Name

The World Wide Name of the port to which the device is attached . It is the 64-bit unique identifier assigned by vendors of FC components such as bridges or native FC devices. Please consult with your FC vendors to find out a port's WWN

Product ID

The product ID of the device. Please consult with your device vendors to determine the product ID.

Discover Devices Supported by TSM

This option discovers devices on a Fibre Channel SAN that are supported by Tivoli Storage Manager and makes them AVAILABLE. If a device is added to or removed from an existing SAN environment, rediscover devices by selecting this option. Devices must be discovered first so that current values of device attributes are shown in the **List Attributes of a Discovered Device** option. Supported devices on FC SAN are tape drives, autochangers, and optical drives. The Tivoli Storage Manager device driver ignores all other device types, such as disk.

Remove All Defined Devices

This option removes all FC SAN-attached Tivoli Storage Manager devices whose state is DEFINED in the ODM database. If necessary, rediscover devices by selecting the **Discover Devices Supported by TSM** option after the removal of all defined devices

Remove a Device

This option removes a single FC SAN-attached Tivoli Storage Manager device whose state is DEFINED in the ODM database. If necessary, rediscover the device by selecting the **Discover Devices Supported by TSM** option after removal of a defined device.

To configure an FC SAN-attached device:

1. Run the SMIT program.
2. Select **Devices**.
3. Select **Tivoli Storage Manager Devices**.
4. Select **Fibre Channel SAN Attached devices**.
5. Select **Discover Devices Supported by TSM**. The discovery process can take some time.
6. Go back to the Fibre Channel menu, and select **List Attributes of a Discovered Device**.
7. Note the 3-character device identifier, which you use when defining a path to the device to Tivoli Storage Manager. For example, if a tape drive has the identifier *mt2*, specify */dev/mt2* as the device name.

Chapter 5. Configuring Storage Devices

This chapter contains concepts and procedures for configuring tape devices, optical disk devices, and removable file devices, such as Iomega Zip drives and Jaz drives.

The following table contains conceptual information needed to understand Tivoli Storage Manager device support:

Concepts:
"Device Configuration Overview" on page 68
"Planning for Devices" on page 68
"Server Storage Options" on page 68
"Defining Devices and Paths" on page 93

Use the following table to locate instructions for specific tasks:

Tasks:
"Configuring SCSI Libraries in a LAN" on page 69
"Configuring SCSI Libraries in a SAN" on page 71
"Configuring IBM 349X Libraries in a LAN" on page 73
"Partitioning IBM 349X Libraries in a LAN" on page 77
"Configuring IBM 349X Libraries in a SAN" on page 81
"Configuring Manually Mounted Devices" on page 84
"Configuring Removable File Devices" on page 86
"Configuring Libraries Controlled by Media Manager Programs" on page 88
"Configuring ACSLS-Managed Libraries" on page 90
"Configuring Tivoli Storage Manager for LAN-free Data Movement" on page 92
"Configuring Tivoli Storage Manager for use with Tivoli Data Protection for NDMP" on page 93

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Note: Some of the tasks described in this chapter require an understanding of Tivoli Storage Manager storage objects. For an introduction to these storage objects, see "Tivoli Storage Manager Storage Objects" on page 42.

© Copyright IBM Corp. 1993, 2002

Device Configuration Overview

Before Tivoli Storage Manager can use a removable media device, you must typically perform the steps described in this section.

1. Plan for the device.

   See “Planning for Devices”.
2. Attach the device to the server system, and ensure that the appropriate device driver is installed and configured. For more information on attaching devices, see “Chapter 4. Attaching Devices to the Server System” on page 57.

   For more information about which device drivers to use, see “Installing and Configuring Device Drivers” on page 59.
3. Define the device to Tivoli Storage Manager.

   Define the library, drive, paths, device class, storage pool, and storage volume objects. For an introduction to these objects, see “Tivoli Storage Manager Storage Objects” on page 42 and “Configuring Devices” on page 48.
4. Define the Tivoli Storage Manager policy that links client data with media for the device.

   Define or update the policy that associates clients with the pool of storage volumes and the device. For an introduction to Tivoli Storage Manager policy, see “How Tivoli Storage Manager Stores Client Data” on page 5. For a description of the default policy, see “The Standard Policy” on page 271.

   Note: As an alternative to creating or modifying a Tivoli Storage Manager policy, you can place the new storage pool in the storage pool migration hierarchy by updating an already defined storage pool.
5. Register clients to the domain associated with the policy that you defined or updated in the preceding step. For more information, see “Chapter 12. Implementing Policies for Client Data” on page 269.
6. Prepare storage volumes for use by the device. At a minimum, you must label volumes for the device. For SCSI, 349X, and ACSLS libraries, add the volumes to the device’s volume inventory by checking in the volumes. For more information, see “Chapter 2. Introducing Storage Devices” on page 29.

Planning for Devices

You should make several operational decisions when adding devices.

1. Determine if the drive is supported by the server. See http://www.tivoli.com/support/storage_mgr/tivolimain.html for more information.
2. Determine if the library is supported by the server. See http://www.tivoli.com/support/storage_mgr/tivolimain.html for more information.
3. Determine the device driver that supports the device.
4. Determine how to attach the devices to the server.

Server Storage Options

Tivoli Storage Manager provides a number of options that you can specify in the server options file (dsmserv.opt) to configure certain server storage operations. The following table provides brief descriptions of these options. See the *Administrator’s Reference* for details.

Table 10. Server Storage Options

Option	Description
3494SHARED	Enables sharing of an IBM 3494 library between a Tivoli Storage Manager server and server applications other than a Tivoli Storage Manager server.
ACSACCESSID	Specifies the ID for the Automatic Cartridge System (ACS) access control.
ACSLOCKDRIVE	Allows the drives within ACSLS libraries to be locked.
ACSQUICKINIT	Allows a quick or full initialization of the ACSLS library.
ACSTIMEOUTX	Specifies the multiple for the built-in timeout value for ACSLS API.
ASSISTVCRRECOVERY	Specifies whether the server assists an IBM 3570 or 3590 drive in recovering from a lost or corrupted Vital Cartridge Records (VCR) condition.
DRIVEACQUIRERETRY	Specifies how many times the server retries the acquisition of a drive in a library when there are no drives available after acquiring a mount point.
ENABLE3590LIBRARY	Enables support for IBM 3590 tape drives in an IBM 349X automated library.
RESOURCETIMEOUT	Specifies how long the server waits for a resource before canceling the pending acquisition of a resource.
SEARCHMPQUEUE	Specifies the order in which the server satisfies requests in the mount queue.

Configuring SCSI Libraries in a LAN

In the following example, an automated SCSI library containing two drives is attached to the server system.

Set up the Device on the Server System

You must first set up the device on the server system. This involves the following tasks:

1. Set the appropriate SCSI ID for each drive and for the library or medium-changer.
2. Physically attach the devices to the server hardware.
3. Install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see “Attaching an Automated Library Device” on page 58 and “Installing and Configuring Device Drivers” on page 59.

Define the Device to Tivoli Storage Manager

1. Define a SCSI library named AUTO8MM. The library type is *SCSI* because the library is a SCSI-controlled automated library. Enter the following command:
   ```
   define library auto8mmlib libtype=scsi
   ```
2. Define a path from the server to the library:
   ```
   define path server1 auto8mmlib srctype=server1 desttype=library
   device=/dev/lb3
   ```

   The DEVICE parameter specifies the device driver’s name for the library, which is the special file name.

   See “Defining Libraries” on page 93 and “SCSI Libraries” on page 30. For more information about paths, see “Defining Paths” on page 95.
3. Define the drives in the library:

```
define drive auto8mmlib drive01 element=82
define drive auto8mmlib drive02 element=83
```

Both drives belong to the AUTO8MMLIB library.

Element address: The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. When you define a drive, the element address is required if more than one drive is in an automated library. The element numbers are taken from the device worksheet filled out in step 7 on page 58. See http://www.tivoli.com/support/storage_mgr/tivolimain.html to determine the element numbers.

See "Defining Drives" on page 94. For more information about paths, see "Defining Paths" on page 95.

4. Define a path from the server to each drive:

   ```
   define path server1 drive01 srctype=server desttype=drive
   library=auto8mmlib device=/dev/mt4
   define path server1 drive02 srctype=server desttype=drive
   library=auto8mmlib device=/dev/mt4
   ```

   The DEVICE parameter specifies the device driver's name for the drive. In this example, each drive is given a device special file name by Tivoli Storage Manager that is unique to the device special file. For more about device special file names, see "Determining Device Special File Names" on page 60. For more information about paths, see "Defining Paths" on page 95.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. We recommend that you use FORMAT=DRIVE as the recording format only if all the drives associated with the device class are identical. For example, to classify two drives in the AUTO8MMLIB library, use the following command to define a device class named AUTO8MM_CLASS:

   ```
   define devclass auto8mm_class library=auto8mmlib devtype=8mm format=drive
   ```

   See "Defining and Updating Device Classes for Tape Devices" on page 143.

6. Verify your definitions by issuing the following commands:

   ```
   query library
   query drive
   query path
   query devclass
   ```

   See "Requesting Information About Libraries" on page 131, "Requesting Information about Drives" on page 133, "Requesting Information about a Device Class" on page 151, and "Requesting Information About Paths" on page 138.

7. Define a storage pool named AUTO8MM_POOL associated with the device class named AUTO8MM_CLASS.

   ```
   define stgpool auto8mm_pool auto8mm_class maxscratch=20
   ```

   Key choices:

   a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping a Client’s Files Together: Collocation” on page 184 and “How Collocation Affects Reclamation” on page 195.

For more information, see “Defining or Updating Primary Storage Pools” on page 158.

Check in and Label Library Volumes

Ensure that enough volumes are available to the server in the library. Keep enough labeled volumes on hand so that you do not run out during an operation such as client backup.

1. Check in the library inventory. The following shows two examples. In both cases, the server uses the name on the barcode label as the volume name.
   - Check in volumes that are already labeled:
     ```
     checkin libvolume auto8mmlib search=yes status=scratch checklabel=barcode
     ```
   - Label and check in volumes:
     ```
     label libvolume auto8mmlib search=yes labelsource=barcode checkin=scratch
     ```
2. Depending on whether you use scratch volumes or private volumes, do one of the following:
   - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
   - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled and checked in. See “Defining Storage Pool Volumes” on page 167.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see “Configuring Policy for Direct-to-Tape Backups” on page 303.
- Have clients back up data to disk. The data is later migrated to tape. For details, see “Overview: The Storage Pool Hierarchy” on page 170.

Configuring SCSI Libraries in a SAN

Using a SAN with Tivoli Storage Manager allows the following functions:

- Multiple Tivoli Storage Manager servers share storage devices.
- Tivoli Storage Manager clients directly access storage devices, both tape libraries and disk storage, that are defined to a Tivoli Storage Manager server (LAN-free data movement). See “Configuring Tivoli Storage Manager for LAN-free Data Movement” on page 92.

The following tasks are required for Tivoli Storage Manager servers to share library devices over a SAN:

1. Set up server-to-server communications.
2. Set up the library on the library manager server. In the example used for this section, the library manager server is named ASTRO.
3. Set up the library on the library client server. In the example used for this section, the library client server is named JUDY.

Setting up Server Communications

Before Tivoli Storage Manager servers can share a storage device over a SAN, you must set up server communications. This requires configuring each server for Enterprise Administration and defining the servers to each other, using the cross-define function. See "Setting Up Communications Among Servers" on page 436 for details. For a discussion regarding the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see "Performing Operations with Shared Libraries" on page 130.

Set up the Device on the Server System

You must first set up the device on the server system. This involves the following tasks:

1. Set the appropriate SCSI ID for each drive and for the library or medium-changer.
2. Physically attach the devices to the server hardware.
3. Install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see "Attaching an Automated Library Device" on page 58 and "Installing and Configuring Device Drivers" on page 59.

Setting up the Library Manager Server

Use the following procedure as an example of how to set up a Tivoli Storage Manager server as a library manager. The server is named ASTRO:

1. Define a shared SCSI library named SANGROUP:

   ```
   define library sangroup libtype=scsi shared=yes
   ```

2. Define a path from the server to the library:

   ```
   define path astro sangroup srctype=server desttype=library
    device=/dev/lb3
   ```

   For more information about paths, see "Defining Paths" on page 95.

3. Define the drives in the library:

   ```
   define drive sangroup drivea element=16
   define drive sangroup driveb element=17
   ```

4. Define a path from the server to each drive:

   ```
   define path astro drivea srctype=server desttype=drive
    library=sangroup device=/dev/rmt4
   define path astro driveb srctype=server desttype=drive
    library=sangroup device=/dev/rmt5
   ```

   For more information about paths, see "Defining Paths" on page 95.

5. Define at least one device class that is associated with the shared library. Set the mount wait times to different values for each server.

   ```
   define devclass tape library=sangroup devtype=3570 mountretention=2 mountwait=15
   ```

6. Check in the library inventory. The following shows two examples. In both cases, the server uses the name on the barcode label as the volume name.

- Check in volumes that are already labeled:

  ```
  checkin libvolume sangroup search=yes status=scratch checklabel=barcode
  ```
- Label and check in volumes:

  ```
  label libvolume sangroup search=yes labelsource=barcode checkin=scratch
  ```

7. Set up a storage pool for the shared library with a maximum of 50 scratch volumes.

   ```
   define stgpool backtape tape maxscratch=50
   ```

Setting up the Library Client Servers

Use the following sample procedure for each Tivoli Storage Manager server that will be a library client server.

1. Define the server that is the library manager:

   ```
   define server astro serverpassword=secret hladdress=9.115.3.45 lladdress=1580
    crossdefine=yes
   ```
2. Define the shared library named SANGROUP, and identify the library manager server's name as the primary library manager. Ensure that the library name agrees with the library name on the library manager:

   ```
   define library sangroup libtype=shared primarylibmanager=astro
   ```
3. Define the drives in the library by using the same names as the drives on the library manager. Element addresses are not required for shared libraries when defining drives on the library client.

   ```
   define drive sangroup drivea
   define drive sangroup driveb
   ```

   Note: We recommend that you define all the drives in the shared library to the library client and library manager servers.
4. Define a path from the library client server to each drive:

   ```
   define path judy drivea srctype=server desttype=drive
    library=sangroup device=/dev/rmt4
   define path judy driveb srctype=server desttype=drive
    library=sangroup device=/dev/rmt5
   ```

   For more information about paths, see "Defining Paths" on page 95.
5. Define at least one device class that is associated with the shared library. Set the mount wait times to different values for each server.

   ```
   define devclass tape library=sangroup devtype=3570 mountretention=2 mountwait=10
   ```
6. Define the storage pool, BACKTAPE, that will use the shared library.

   ```
   define stgpool backtape tape maxscratch=50
   ```

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 303.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 170.

Configuring IBM 349X Libraries in a LAN

In the following example, an IBM 3494 library containing two drives is attached to the server system.

Set up the Device on the Server System

You must first set up the 349X library on the server system. This involves the following tasks:

1. Set the 349X Library Manager Control Point, or LMCP. This procedure is described in the *IBM SCSI Tape Drive, Medium Changer, and Library Device Drivers Installation and User's Guide* for AIX.
2. Physically attach the devices to the server hardware.
3. Install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see "Attaching an Automated Library Device" on page 58 and "Installing and Configuring Device Drivers" on page 59.

Define the Device to Tivoli Storage Manager

1. Define a 349X library named 3494LIB:

   ```
   define library 3494lib libtype=349x
   ```

2. Define a path from the server to the library:

   ```
   define path server1 3494lib srctype=server desttype=library
   device=/dev/lmcp0,/dev/lmcp1
   ```

   The DEVICE parameter specifies the device special files for the LMCP.

   See "Defining Libraries" on page 93 and "SCSI Libraries" on page 30. For more information about paths, see "Defining Paths" on page 95.

3. Define the drives in the library:

   ```
   define drive 3494lib drive01
   define drive 3494lib drive02
   ```

   Both drives belong to the 3494LIB library.

   See "Defining Drives" on page 94.

4. Define a path from the server to each drive:

   ```
   define path server1 drive01 srctype=server desttype=drive
   library=3494lib device=/dev/rmt0
   define path server1 drive02 srctype=server desttype=drive
   library=3494lib device=/dev/rmt1
   ```

   The DEVICE parameter gives the device special file for the drive. In this example, each drive is given a device special file name by Tivoli Storage Manager that is unique to the device special file. For more about device names, see "Determining Device Special File Names" on page 60. For more information about paths, see "Defining Paths" on page 95.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. We recommend that you use FORMAT=DRIVE as the recording format only if all the drives associated with the device class are identical. For example, to classify the two drives in the 3494LIB library, use the following command to define a device class named 3494_CLASS:

   ```
   define devclass 3494_class library=3494lib devtype=3590 format=drive
   ```

   See "Defining and Updating Device Classes for Tape Devices" on page 143.

6. Verify your definitions by issuing the following commands:

```
query library
query drive
query path
query devclass
```

See “Requesting Information About Libraries” on page 131, “Requesting Information about Drives” on page 133, “Requesting Information about a Device Class” on page 151, and “Requesting Information About Paths” on page 138.

7. Define a storage pool named 3494_POOL associated with the device class named 3494_CLASS.

   ```
   define stgpool 3494_pool 3494_class maxscratch=20
   ```

   Key choices:

   a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
   b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping a Client’s Files Together: Collocation” on page 184 and “How Collocation Affects Reclamation” on page 195.

   For more information, see “Defining or Updating Primary Storage Pools” on page 158.

Categories in an IBM 349X Automated Library

The 349X library manager tracks the category number of each volume in the library. A single category number identifies all volumes used for the same purpose or application. These category numbers are useful when multiple systems share the resources of a single library.

Attention: If other systems or other Tivoli Storage Manager servers connect to the same 349X library, each must use a unique set of category numbers. Otherwise, two or more systems may try to use the same volume, and cause a corruption or loss of data.

Typically, a software application that uses a 349X library device uses volumes in one or more categories that are reserved for that application. To avoid loss of data, each application sharing the library must have unique categories. When you define a 349X library to the server, you can use the PRIVATECATEGORY and SCRATCHCATEGORY parameters to specify the category numbers for private and scratch Tivoli Storage Manager volumes in that library. See “Tivoli Storage Manager Volumes” on page 41 for more information on private and scratch volumes.

When a volume is first inserted into the library, either manually or automatically at the convenience I/O station, the volume is assigned to the insert category (X'FF00'). A software application, such as Tivoli Storage Manager, can contact the library manager to change a volume’s category number. For Tivoli Storage

Manager, you would use the CHECKIN LIBVOLUME command (see “Checking New Volumes into a Library” on page 117).

The number of categories that the server requires depends on whether you have enabled support for 3590 drives. If support is not enabled for 3590 drives, the server reserves two category numbers in each 349X library that it accesses: one for private volumes and one for scratch volumes. If you enable 3590 support, the server reserves three categories in the 349X library: private, scratch for 3490 drives, and scratch for 3590 drives.

The default values for the PRIVATECATEGORY and SCRATCHCATEGORY parameters are the same as when 3590 support is not enabled. However, the server automatically creates the scratch category for 3590 drives, by adding the number 1 to the SCRATCHCATEGORY value you specify. For example, suppose you enter the following command:

```
define library my3494 libtype=349x privatecategory=400 scratchcategory=401
```

Tivoli Storage Manager uses the following categories in the library:

- **400 (X'190')** Private volumes (for both 3490 and 3590 drives)
- **401 (X'191')** Scratch volumes for 3490 drives
- **402 (X'192')** Scratch volumes for 3590 drives

To avoid overlapping categories, ensure that the value specifies for the private category is not equal to the scratch category value plus 1.

Attention: The default values for the categories may be acceptable in most cases. However, if you connect other systems or Tivoli Storage Manager servers to a single 349X library, ensure that each uses unique category numbers. Otherwise, two or more systems may try to use the same volume, and cause a corruption or loss of data.

Also, if you share a 349X library with other Tivoli Storage Manager servers or other applications or systems, be careful when enabling 3590 support to prevent loss of data. See “Enabling Support for IBM 3590 Drives in Existing 349X Libraries”. For a discussion regarding the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see “Performing Operations with Shared Libraries” on page 130.

Enabling Support for IBM 3590 Drives in Existing 349X Libraries

The new category that the server creates for 3590 scratch volumes can duplicate a category already assigned to another application and cause loss of data. If you are currently sharing a 349X library with other Tivoli Storage Manager servers or other applications or systems and you enable support for 3590 drives, you need to be careful. The server automatically creates a third category for 3590 scratch volumes by adding one to the existing scratch category for any 349X libraries defined to Tivoli Storage Manager.

To prevent loss of data, do one of the following before enabling 3590 support:

- Update other applications and systems to ensure that there is no conflicting use of category numbers.
- Delete the existing library definition and then define it again using a new set of category numbers that do not conflict with categories used by other systems or applications using the library. Do the following:

1. Use an external utility (such as mtlib) to reset all of the Tivoli Storage Manager volumes to the insert category.
2. Delete the 349X library definition.
3. Define the 349X library again, using new category numbers.

   Check in the Tivoli Storage Manager volumes that you put in the insert category in step 1.
4. Specify the volume type. Users with both 3490 and 3590 drives must specify DEVTYPE=3590 for a 3590 volume type.

For more information about checking in volumes, see "Checking New Volumes into a Library" on page 117.

After taking steps to prevent data loss, enable 3590 support by adding the following line to the server options file (dsmserv.opt):

```
ENABLE3590LIBRARY  YES
```

Stop and start the server to make this change effective.

Check in and Label a Library Volume

Ensure that enough volumes are available to the server in the library. Keep enough labeled volumes on hand so that you do not run out during an operation such as client backup.

1. Check in the library inventory. The following shows two examples.
   - Check in volumes that are already labeled:

     ```
     checkin libvolume 3494lib search=yes status=scratch checklabel=no
     ```
   - Label and check in volumes:

     ```
     label libvolume 3494lib search=yes checkin=scratch
     ```
2. Depending on whether you use scratch volumes or private volumes, do one of the following:
   - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
   - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled and checked in. See "Defining Storage Pool Volumes" on page 167.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 303.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 170.

Partitioning IBM 349X Libraries in a LAN

Tivoli Storage Manager uses the capability of the 349X library manager, which allows you to partition libraries that reside on a LAN between multiple Tivoli Storage Manager server applications. Library partitioning differs from library sharing in that there are no Tivoli Storage Manager library managers or clients.

Each server has the same access to the library. Each drive that resides in the library is assigned to one server. Each server can then only access those drives it has been assigned. Tape volume categories must also be assigned to each server that has access to the library. As a result, library partitioning does not allow dynamic sharing of drives or tape volumes since they are pre-assigned to different servers using different names and category codes.

In the following example, an IBM 3494 library containing four drives is attached to a Tivoli Storage Manager server system named ASTRO and to another Tivoli Storage Manager server system named JUDY.

Set up the Device on the Server System

You must first set up the 349X library on the server system. This involves the following tasks:

1. Set the 349X Library Manager Control Point, or LMCP. This procedure is described in the *IBM SCSI Tape Drive, Medium Changer, and Library Device Drivers Installation and User's Guide* for AIX.
2. Physically attach the devices to the server hardware.
3. Install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see "Attaching an Automated Library Device" on page 58 and "Installing and Configuring Device Drivers" on page 59.

Define the Devices to Tivoli Storage Manager ASTRO

1. Define the 3494 library named 3494LIB:

   ```
   define library 3494lib libtype=3494 privatecategory=400 scratchcategory=600
   ```

   The PRIVATECATEROY and SCRATCHCATEGORY are set differently from the default settings. See "Categories in an IBM 349X Automated Library" on page 75.
2. Define the path from the server, ASTRO, to the library:

   ```
   define path astro 3494lib srctype=server desttype=library
    device=/dev/lmcp0
   ```

   The DEVICE parameter specifies the device special file for the LMCP.

   See "Defining Libraries" on page 93 and "SCSI Libraries" on page 30. For more information about paths, see "Defining Paths" on page 95.
3. Define the drives that are partitioned to server ASTRO:

   ```
   define drive 3494lib drive1
   define drive 3494lib drive2
   ```
4. Define the path from the server, ASTRO, to each of the drives:

   ```
   define path astro drive1 srctype=server desttype=drive library=3494lib
    device=/dev/rmt0
   define path astro drive2 srctype=server desttype=drive library=3494lib
    device=/dev/rmt1
   ```

   The DEVICE parameter specifies the device special file for the drive. In this example, each drive is given a Tivoli Storage Manager name that is unique to the device special file. For more about device names, see "Determining Device Special File Names" on page 60. For more information about paths, see "Defining Paths" on page 95.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. We recommend that you use FORMAT=DRIVE as the recording format only if all the drives associated with the device class are identical. For example, to classify the two drives in the 3494LIB library, use the following command to define a device class named 3494_CLASS:

   ```
   define devclass 3494_class library=3494lib devtype=3590 format=drive
   ```

   See “Defining and Updating Device Classes for Tape Devices” on page 143.
6. Verify your definitions by issuing the following commands:

   ```
   query library
   query drive
   query path
   query devclass
   ```

   See “Requesting Information About Libraries” on page 131, “Requesting Information about Drives” on page 133, “Requesting Information about a Device Class” on page 151, and “Requesting Information About Paths” on page 138.
7. Define a storage pool named 3494_POOL associated with the device class named 3494_CLASS.

   ```
   define stgpool 3494_pool 3494_class maxscratch=20
   ```

   Key choices:

   a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.
   b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping a Client’s Files Together: Collocation” on page 184 and “How Collocation Affects Reclamation” on page 195.

   For more information, see “Defining or Updating Primary Storage Pools” on page 158.

Define the Devices to Tivoli Storage Manager JUDY

1. Define the 3494 library named 3494LIB:

   ```
   define library 3494lib libtype=3494 privatecategory=112 scratchcategory=300
   ```

   The privatecategory and scratchcategory are defined differently that the first server’s definition. See “Categories in an IBM 349X Automated Library” on page 75.
2. Define the path from the server, JUDY, to the library:

   ```
   define path judy 3494lib srctype=server desttype=library
   device=/dev/lmcp0
   ```

   The DEVICE parameter specifies the device special files for the LMCP.

See “Defining Libraries” on page 93 and “SCSI Libraries” on page 30. For more information about paths, see “Defining Paths” on page 95.

3. Define the drives that are partitioned to server JUDY:

   ```
   define drive 3494lib drive3
   define drive 3494lib drive4
   ```

4. Define the path from the server, JUDY, to each of the drives:

   ```
   define path judy drive3 srctype=server desttype=drive library=3494lib
    device=/dev/rmt2
   define path judy drive4 srctype=server desttype=drive library=3494lib
    device=/dev/rmt3
   ```

   The DEVICE parameter gives the device special file for the drive. In this example, each drive is given a Tivoli Storage Manager name that is unique to the device special file. For more about device names, see “Determining Device Special File Names” on page 60. For more information about paths, see “Defining Paths” on page 95.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. We recommend that you use FORMAT=DRIVE as the recording format only if all the drives associated with the device class are identical. For example, to classify the two drives in the 3494LIB library, use the following command to define a device class named 3494_CLASS:

   ```
   define devclass 3494_class library=3494lib devtype=3590 format=drive
   ```

   See “Defining and Updating Device Classes for Tape Devices” on page 143.

6. Verify your definitions by issuing the following commands:

   ```
   query library
   query drive
   query path
   query devclass
   ```

   See “Requesting Information About Libraries” on page 131, “Requesting Information about Drives” on page 133, and “Requesting Information about a Device Class” on page 151.

7. Define a storage pool named 3494_POOL associated with the device class named 3494_CLASS.

   ```
   define stgpool 3494_pool 3494_class maxscratch=20
   ```

   Key choices:

   a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

   b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping a Client’s Files Together: Collocation” on page 184 and “How Collocation Affects Reclamation” on page 195.

   For more information, see “Defining or Updating Primary Storage Pools” on page 158.

Configuring IBM 349X Libraries in a SAN

Using a SAN with Tivoli Storage Manager allows the following functions:

- Multiple Tivoli Storage Manager servers share storage devices.
- Tivoli Storage Manager clients directly access storage devices, both tape libraries and disk storage, that are defined to a Tivoli Storage Manager server (LAN-free data movement). See “Configuring Tivoli Storage Manager for LAN-free Data Movement” on page 92.

Note: Tivoli Storage Manager supports library sharing and LAN-free features with IBM 349X libraries. If you are already using an IBM 349X library with Tivoli Storage Manager in a non-SAN environment, see “Migrating an IBM 349X Library to SAN Support” on page 83.

The following tasks are required for Tivoli Storage Manager servers to share library devices over a SAN:

1. Set up server-to-server communications.
2. Set up the library on the library manager server. In the following example, the library manager server is named MANAGER.
3. Set up the library on the library client server. In the following example, the library client server is named CLIENT.

See “Categories in an IBM 349X Automated Library” on page 75 and “Enabling Support for IBM 3590 Drives in Existing 349X Libraries” on page 76 for additional information about configuring 349X libraries.

Setting up Server Communications

Before Tivoli Storage Manager servers can share a storage device over a SAN, you must set up server communications. This requires configuring each server for Enterprise Administration and defining the servers to each other, using the cross-define function. See “Setting Up Communications Among Servers” on page 436 for details. For a discussion regarding the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see “Performing Operations with Shared Libraries” on page 130.

Set up the Device on the Server System

You must first set up the device on the server system. This involves the following tasks:

1. Set the 349X Library Manager Control Point (LMCP). This procedure is described in the *IBM SCSI Tape Drive, Medium Changer, and Library Device Drivers Installation and User’s Guide* for AIX.
2. Physically attach the devices to the server hardware.
3. Install and configure the appropriate device drivers for the devices.
4. Determine the device names that are needed to define the devices to Tivoli Storage Manager.

For details, see “Attaching an Automated Library Device” on page 58 and “Installing and Configuring Device Drivers” on page 59.

Setting up the Library Manager Server

Use the following procedure as an example of how to set up a Tivoli Storage Manager server as a library manager named MANAGER:

1. Define a 349X library named 3494SAN:

```
define library 3494san libtype=349x shared=yes
```

2. Define a path from the server to the library:

   ```
   define path manager 3494san srctype=server desttype=library
    device=/dev/lmcp0,/dev/lmcp1
   ```

   For more information about paths, see “Defining Paths” on page 95.
3. Define the drives in the library:

   ```
   define drive 3494san drivea
   define drive 3494san driveb
   ```
4. Define a path from the server to each drive:

   ```
   define path manager drivea srctype=server desttype=drive library=3494san
    device=/dev/rmt0
   define path manager driveb srctype=server desttype=drive library=3494san
    device=/dev/rmt1
   ```

   For more information about paths, see “Defining Paths” on page 95.
5. Define at least one device class associated with the shared library. Set the mount wait times to different values for each server.

   ```
   define devclass 3494_class library=3494san devtype=3590 mountretention=2 mountwait=10
   ```
6. Check in the library inventory. The following shows two examples. In both cases, the server uses the name on the barcode label as the volume name.

   To check in volumes that are already labeled, use the following command:

   ```
   checkin libvolume 3494san search=yes status=scratch checklabel=no
   ```

   To label and check in the volumes, use the following command:

   ```
   label libvolume 3494san checkin=scratch search=yes
   ```
7. Set up a storage pool for the shared library with a maximum of 50 scratch volumes.

   ```
   define stgpool 3494_sanpool tape maxscratch=50
   ```

Setting up the Library Client Servers

Use the following sample procedure for each Tivoli Storage Manager server that will be a library client server.

1. Define the server that is the library manager:

   ```
   define server manager serverpassword=secret hladdress=9.115.3.45 lladdress=1580
    crossdefine=yes
   ```
2. Define a shared library named 3494SAN, and identify the library manager:

   Note: Ensure that the library name agrees with the library name on the library manager.

   ```
   define library 3494san libtype=shared primarylibmanager=manager
   ```
3. Define the drives in the library by using the same names as the drives on the library manager. Element addresses are not required for shared libraries when defining drives on the library client.

   ```
   define drive 3494san drivea
   define drive 3494san driveb
   ```

   Note: We recommend that you define all the drives in the shared library to both the library client and library manager servers.
4. Define a path from the library client server to each drive:

```
define path client drivea srctype=server desttype=drive
 library=3494san device=/dev/rmt0
define path client driveb srctype=server desttype=drive
 library=3494san device=/dev/rmt1
```

For more information about paths, see “Defining Paths” on page 95.

5. Define at least one device class associated with the shared library. Set the mount wait times to different values for each server.

   ```
   define devclass 3494_class library=3494san devtype=3590 mountretention=2 mountwait=15
   ```

6. Define the storage pool, BACKTAPE, that will use the shared library.

   ```
   define stgpool backtape 3494_class maxscratch=50
   ```

7. Repeat this procedure to define additional servers as library clients. For a discussion regarding the interaction between library clients and the library manager in processing Tivoli Storage Manager operations, see “Performing Operations with Shared Libraries” on page 130.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see “Configuring Policy for Direct-to-Tape Backups” on page 303.
- Have clients back up data to disk. The data is later migrated to tape. For details, see “Overview: The Storage Pool Hierarchy” on page 170.

Migrating an IBM 349X Library to SAN Support

If you are already using an IBM 349X library with Tivoli Storage Manager in a non-SAN environment, you will have to perform the following procedure to migrate and correctly configure the library for a SAN environment. You must ensure that the 3494SHARED option is disabled in the dsmserv.opt file:

1. Do the following on each server sharing a 3494 library:
   a. Update the storage pools using the UPDATE STGPOOL command. Set the value for the HIGHMIG and LOWMIG parameters to 100%.
   b. Stop the server by issuing the HALT command or accessing the TSM Console and clicking **Stop** for the server.
   c. Edit the dsmserv.opt file:
      1) Comment out the 3494SHARED YES option line
      2) Activate the **disablescheds yes** option line if it is not active
      3) Activate the **expinterval** *x* option line if it is not active and change the value to 0; **expinterval 0**
   d. Start the server.
   e. Enter the following Tivoli Storage Manager command:

      ```
      disable sessions
      ```
2. Set up the library manager on a Tivoli Storage Manager server of your choosing.
3. Do the following on the remaining Tivoli Storage Manager servers:
   a. Save the volume history file.
   b. Check out all the volumes in the library inventory. Use the CHECKOUT LIBVOLUME command with REMOVE=NO.
   c. Follow the library client setup procedure.
4. Do the following on the library manager server:

a. Check in each library client's volumes. Use the CHECKIN LIBVOLUME command with the following parameter settings:
   - STATUS=PRIVATE
   - OWNER=<*library client name*>

   Note: You can use the saved volume history files from the library clients as a guide.

b. Check in any remaining volumes as scratch volumes. Use the CHECKIN LIBVOLUME command with STATUS=SCRATCH.

5. Halt all the servers.
6. Edit the dsmserv.opt file and comment out the **disablescheds yes** and **expinterval 0** option lines.
7. Start the servers.

Configuring Manually Mounted Devices

In the following example, two 8mm drives are attached to the server system. Because an operator must mount tapes for these drives, you must define the drives as part of a *manual* library.

Set up the Device on the Server System

You must first set up the device on the server system. This involves the following tasks:

1. Set the appropriate SCSI ID for the device.
2. Physically attach the device to the server hardware.
3. Install and configure the appropriate device driver for the device.
4. Determine the device name that is needed to define the device to Tivoli Storage Manager.

See "Attaching a Manual Drive" on page 57 and "Installing and Configuring Device Drivers" on page 59 for details.

Define the Device to Tivoli Storage Manager

1. Define a manual library named MANUAL8MM:
   ```
   define library manual8mm libtype=manual
   ```
2. Define the drives in the library:
   ```
   define drive manual8mm drive01
   define drive manual8mm drive02
   ```
   See "Defining Drives" on page 94 and http://www.tivoli.com/support/storage_mgr/tivolimain.html.
3. Define a path from the server to each drive:
   ```
   define path server1 drive01 srctype=server desttype=drive
   library=manual8mm device=/dev/mt1
   define path server1 drive02 srctype=server desttype=drive
   library=manual8mm device=/dev/mt2
   ```
   In this example, the path between the server, SERVER1, and the drive called DRIVE01 is defined using the special file name */dev/mt1*. The path between the server, SERVER1, and the drive DRIVE02 is defined using the special file name

/dev/mt2. You might prefer to have the device special file name and the TSM name match. For more about device names, see "Determining Device Special File Names" on page 60.

For more information about paths, see "Defining Paths" on page 95.

4. Classify the drives according to type by defining a device class named TAPE8MM_CLASS. We recommend that you use FORMAT=DRIVE as the recording format only if all the drives associated with the device class are identical.

   ```
   define devclass tape8mm_class library=manual8mm devtype=8mm format=drive
   ```

   A closer look: When you associate more than one drive to a single device class through a manual library, ensure that the recording formats and media types of the devices are compatible. If you have a 4mm tape drive and an 8mm tape drive, you must define separate manual libraries and device classes for each drive.

   See "Defining and Updating Device Classes for Tape Devices" on page 143.

5. Verify your definitions by issuing the following commands:

   ```
   query library
   query drive
   query path
   query devclass
   ```

   See "Requesting Information About Libraries" on page 131, "Requesting Information about Drives" on page 133, "Requesting Information about a Device Class" on page 151, and "Requesting Information About Paths" on page 138 .

6. Define a storage pool named TAPE8MM_POOL associated with the device class named TAPE8MM_CLASS:

   ```
   define stgpool tape8mm_pool tape8mm_class maxscratch=20
   ```

   Key choices:

   a. Scratch volumes are empty volumes that are labeled and available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can use any scratch volumes available without further action on your part. If you do not allow scratch volumes (MAXSCRATCH=0), you must perform the extra step of explicitly defining each volume to be used in the storage pool.

   b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see "Keeping a Client's Files Together: Collocation" on page 184 and "How Collocation Affects Reclamation" on page 195.

   See "Defining or Updating Primary Storage Pools" on page 158.

Label Volumes

Use the following procedure to ensure that volumes are available to the server:

1. Label volumes that do not already have a standard label. For example, enter the following command to use one of the drives to label a volume with the ID of vol001:

   ```
   label libvolume manual8mm vol001
   ```

   Keep enough labeled volumes on hand so that you do not run out during an operation such as client backup.
2. Depending on whether you use scratch volumes or private volumes, do one of the following:
   - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
   - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled. For information on defining volumes, see “Defining Storage Pool Volumes” on page 167.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see “Configuring Policy for Direct-to-Tape Backups” on page 303.
- Have clients back up data to disk. The data is later migrated to tape. For details, see “Overview: The Storage Pool Hierarchy” on page 170.

Configuring Removable File Devices

Removable file support includes Iomega ZIP drives and JAZ drives, and CD-ROM drives.

Support for removable file devices allows portability of media between UNIX systems. It also allows this media to be used to transfer data between systems that support the media. Removable file support allows the server to read data from a FILE device class that is copied to removable file media through third-party software. The media is then usable as input media on a target Tivoli Storage Manager server that uses the REMOVABLEFILE device class for input.

Note: Software for writing CD-ROMs may not work consistently across platforms.

Use a MAXCAPACITY value that is less than one CD-ROM’s usable space to allow for a one-to-one match between files from the FILE device class and copies that are on CD-ROM. Use the DEFINE DEVCLASS or UPDATE DEVCLASS commands to set the MAXCAPACITY parameter of the FILE device class to a value less than 650MB.

Example of Removable File Support

Use these steps as an example of Tivoli Storage Manager REMOVABLEFILE (CD-ROM) support. This example takes an export object and moves it from one server to another.

Server A

1. Define a device class with a device type of FILE.

```
define devclass file devtype=file directory=/home/user1
```

2. Export the node. This command results in a file name */home/user1/CDR03* that contains the export data for node USER1.

   ```
   export node user1 filedata=all devclass=file vol=cdr03
   ```

   You can use software for writing CD-ROMs to create a CD with volume label CDR03 that contains a single file that is also named CDR03.

Server B

1. Follow the manufacturer's instructions to attach the device to your server.
2. Issue this command on your system to mount the CD-ROM.

   ```
   mount -r -v cdrfs /dev/cd0 /cdrom
   ```

   -r Specifies a read-only file system

   –v cdrfs
   Specifies that the media has a CD file system

   /dev/cd0
   Specifies the physical description of the first CD-ROM on the system

   /cdrom
   Specifies the mount point of the first CD-ROM drive

   Note: CD-ROM drives lock while the file system is mounted. This prevents use of the eject button on the drive.
3. Ensure that the media is labeled. The software that you use for making a CD also labels the CD. Before you define the drive, you must put formatted, labeled media in the drive. For label requirements, see "Labeling Requirements for Optical and Other Removable Files Devices". When you define the drive, the server verifies that a valid file system is present.
4. Define a manual library named CDROM:

   ```
   define library cdrom libtype=manual
   ```
5. Define the drive in the library:

   ```
   define drive cdrom cddrive
   ```
6. Define a path from the server to the drive at mount point */cdrom*:

   ```
   define path serverb cddrive srctype=server desttype=drive
   library=cdrom device=/cdrom
   ```

   For more information about paths, see "Defining Paths" on page 95.
7. Define a device class with a device type of REMOVABLEFILE. The device type must be REMOVABLEFILE.

   ```
   define devclass cdrom devtype=removablefile library=cdrom
   ```
8. Issue the following Tivoli Storage Manager command to import the node data on the CD-ROM volume CDR03.

   ```
   import node user1 filedata=all devclass=cdrom vol=cdr03
   ```

Labeling Requirements for Optical and Other Removable Files Devices

Tivoli Storage Manager does not provide utilities to format or label media for the REMOVABLEFILE device type. You must use another application to copy the FILE device class data from the CD-ROM as a file that has the same name as the volume label. The software used to copy the FILE device class data must also label the removable media.

The label on the media must meet the following restrictions:

- No more than 11 characters
- No embedded blanks or periods
- File name must be the same as the volume label

Configuring Libraries Controlled by Media Manager Programs

You can use an external media manager program with Tivoli Storage Manager to manage your removable media. While the server tracks and manages client data, the media manager, operating entirely outside of the I/O data stream, labels, catalogs, and tracks physical volumes. The media manager also controls library drives, slots, and doors.

A programming interface is provided that lets you use a variety of media managers. See "Appendix A. External Media Management Interface Description" on page 603 for a complete description of this interface. See "Setting up Tivoli Storage Manager to Work with an External Media Manager" for setup procedures.

To use a media manager with Tivoli Storage Manager, define a library that has a library type of EXTERNAL. The library definition will point to the media manager rather than a physical device.

Setting up Tivoli Storage Manager to Work with an External Media Manager

To use the External Media Management Interface with a media manager, do the following procedure. This example is for an 8mm medium-changer device containing two drives.

1. Set up the media manager to interface with Tivoli Storage Manager. For more information, see "Appendix A. External Media Management Interface Description" on page 603.
2. Define an external library named MEDIAMGR:

   ```
   define library mediamgr libtype=external
   ```

   Note: You do not define the drives to the server in an externally managed library.
3. Define a path from the server to the library:

   ```
   define path server1 mediamgr srctype=server desttype=library
   externalmanager=/u/server/mediamanager
   ```

   In the EXTERNALMANAGER parameter, specify the media manager's installed path. For more information about paths, see "Defining Paths" on page 95.
4. Define device class, EXTCLASS, for the library with a device type of 8mm:

   ```
   define devclass extclass library=mediamgr mountretention=5 mountlimit=2
   ```

   The MOUNTLIMIT parameter specifies the number of drives in the library device.

   Notes:

   a. For environments in which devices are shared across storage applications, the MOUNTRETENTION setting should be carefully considered. This parameter determines how long an idle volume remains in a drive. Because some media managers will not dismount an allocated drive to satisfy pending requests, you might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance.

   b. It is recommended that you explicitly specify the mount limit instead of using MOUNTLIMIT=DRIVES.

5. Define a storage pool, EXTPOOL, for the device class. For example:

   ```
   define stgpool extpool extclass maxscratch=500
   ```

   Key choices:

   a. Scratch volumes are labeled, empty volumes that are available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

   b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see “Keeping a Client’s Files Together: Collocation” on page 184 and “How Collocation Affects Reclamation” on page 195.

Managing Externally Controlled Tivoli Storage Manager Media

Refer to the documentation for the media manager for detailed setup and management information. The following are some Tivoli Storage Manager-specific issues that you should consider:

Labeling Media

The media manager handles the labeling of media. However, you must ensure that an adequate supply of blank media is available.

Checking Media into the Library

Externally managed media is not tracked in the Tivoli Storage Manager volume inventory. Therefore, you will not perform library check-in procedures.

Using Tivoli Disaster Recovery Manager

If you are using DRM, you can use the MOVE DRMEDIA command to request the removal of media from the library. For more information, see “Chapter 23. Using Tivoli Disaster Recovery Manager” on page 545.

Migrating Media to External Media Manager Control

We strongly recommend that you not migrate media from Tivoli Storage Manager control to control by an external media manager. Instead, use external media management on a new Tivoli Storage Manager configuration or when defining externally managed devices to the server.

Deleting Tivoli Storage Manager Storage Pools from Externally Managed Libraries

Before deleting externally managed storage pools, first delete any volumes associated with the Tivoli Storage Manager library. For more information, see “Deleting a Storage Pool Volume with Data” on page 222.

Troubleshooting Database Errors

Error conditions can cause the Tivoli Storage Manager volume database and the media manager’s volume database to become unsynchronized. The most likely symptom of this problem is that the volumes in the media manager’s database are not known to the server, and thus not available for use. Verify the Tivoli Storage Manager volume list and any disaster recovery media. If volumes not identified to the server are found, use the media manager interface to deallocate and delete the volumes.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see "Configuring Policy for Direct-to-Tape Backups" on page 303.
- Have clients back up data to disk. The data is later migrated to tape. For details, see "Overview: The Storage Pool Hierarchy" on page 170.

Configuring ACSLS-Managed Libraries

Tivoli Storage Manager supports tape libraries controlled by StorageTek Automated Cartridge System Library Software (ACSLS). The ACSLS library server manages the physical aspects of tape cartridge storage and retrieval. The ACSLS client application communicates with the library server to access tape cartridges in an automated library. Tivoli Storage Manager is one of the applications that gains access to tape cartridges by interacting with ACSLS through its client, which is known as the control path. The server reads and writes data on tape cartridges by interacting directly with tape drives through the data path. The control path and the data path are two different paths. The ACSLS client daemon must be initalized before starting the server. See */usr/tivoli/tsm/devices/bin/rc.acs_ssi* for the client daemon invocation. For detailed installation, configuration, and system administration of ACSLS, refer to the appropriate StorageTek documentation.

Set up the Device on the Server System

The library is attached to the ACSLS server, and the drives are attached to the Tivoli Storage Manager server. The ACSLS server and the Tivoli Storage Manager server must be on different systems. Refer to the ACSLS installation documentation for details about how to set up the library.

Define the Device to Tivoli Storage Manager

1. Define an ACSLS library named ACSLIB:

   ```
   define library acslib libtype=acsls acsid=1
   ```

   The parameter ACSID specifies the number that Automatic Cartridge System System Administrator (ACSSA) assigned to the library. Issue QUERY ACS in your system to determine the number for your library ID.
2. Define a path from the server to the library:

   ```
   define path server1 acslib srctype=server desttype=library acsid=1
   ```

   For more information about paths, see "Defining Paths" on page 95.
3. Define the drives in the library:

   ```
   define drive acslib drive01 acsdrvid=1,2,3,4
   define drive acslib drive02 acsdrvid=1,2,3,5
   ```

   The ACSDRVID parameter specifies the ID of the drive that is being accessed. The drive ID is a set of numbesr that indicate the physical location of a drive within an ACSLS library. This drive ID must be specified as *a*, *l*, *p*, *d*, where *a* is the ACSID, *l* is the LSM (library storage module), *p* is the panel number, and *d* is the drive ID. The server needs the drive ID to connect the physical location ofthe drive to the drive's SCSI address. See the Storage Tek documentation for details.

See "Defining Drives" on page 94.

4. Define a path from the server to each drive:

   ```
   define path server1 drive01 srctype=server desttype=drive
   library=acslib device=/dv/mt0
   define path server1 drive02 srctype=server desttype=drive
   library=acslib device=/dv/mt1
   ```

   The DEVICE parameter specifies the device special file for the drive. In this example, each drive is given a Tivoli Storage Manager name that is unique to the device special file. For more about device names, see "Determining Device Special File Names" on page 60. For more information about paths, see "Defining Paths" on page 95.

5. Classify drives according to type by defining Tivoli Storage Manager device classes. We recommend that you use FORMAT=DRIVE as the recording format only if all the drives associated with the device class are identical. For example, to classify the two drives in the ACSLIB library, use the following command to define a device class named ACS_CLASS:

   ```
   define devclass acs_class library=acslib devtype=ecartridge format=drive
   ```

   See "Defining and Updating Device Classes for Tape Devices" on page 143.

6. To check what you have defined, enter the following commands:

   ```
   query library
   query drive
   query path
   query devclass
   ```

   See "Requesting Information About Libraries" on page 131, "Requesting Information about Drives" on page 133, "Requesting Information about a Device Class" on page 151, and "Requesting Information About Paths" on page 138.

7. Create the storage pool to use the devices in the device class you just defined. For example, define a storage pool named ACS_POOL associated with the device class ACS_CLASS:

   ```
   define stgpool acs_pool acs_class maxscratch=20
   ```

   Key choices:

   a. Scratch volumes are labeled, empty volumes that are available for use. If you allow scratch volumes for the storage pool by specifying a value for the maximum number of scratch volumes, the server can choose from the scratch volumes available in the library, without further action on your part. If you do not allow scratch volumes, you must perform the extra step of explicitly defining each volume to be used in the storage pool.

   b. Collocation is turned off by default. Collocation is a process by which the server attempts to keep all files belonging to a client node or client file space on a minimal number of volumes. Once clients begin storing data in a storage pool with collocation off, you cannot easily change the data in the storage pool so that it is collocated. To understand the advantages and disadvantages of collocation, see "Keeping a Client's Files Together: Collocation" on page 184 and "How Collocation Affects Reclamation" on page 195.

For more information, see “Defining or Updating Primary Storage Pools” on page 158 .

Label and Check In a Library Volume

Ensure that enough volumes are available to the server in the library. You must label volumes that do not already have a standard label. Keep enough labeled volumes on hand so that you do not run out during an operation such as client backup.

1. Check in the library inventory. The following shows two examples.
   - Check in volumes that are already labeled:
     ```
     checkin libvolume acslib search=yes status=scratch checklabel=no
     ```
   - Label and check in volumes:
     ```
     label libvolume acslib search=yes overwrite=no checkin=scratch
     ```
2. Depending on whether you use scratch volumes or private volumes, do one of the following:
   - If you use only scratch volumes, ensure that enough scratch volumes are available. For example, you may need to label more volumes. As volumes are used, you may also need to increase the number of scratch volumes allowed in the storage pool that you defined for this library.
   - If you want to use private volumes in addition to or instead of scratch volumes in the library, define volumes to the storage pool you defined. The volumes you define must have been already labeled and checked in. See “Defining Storage Pool Volumes” on page 167.

Using the Devices to Store Client Data

After you have attached and defined your devices, you can store client data in two ways:

- Have clients back up data directly to tape. For details, see “Configuring Policy for Direct-to-Tape Backups” on page 303.
- Have clients back up data to disk. The data is later migrated to tape. For details, see “Overview: The Storage Pool Hierarchy” on page 170.

Configuring Tivoli Storage Manager for LAN-free Data Movement

You are able to configure the Tivoli Storage Manager client and server so that the client, through a storage agent, can move its data directly to storage on a SAN. This function, called LAN-free data movement, is provided by the Managed System for SAN feature. As part of the configuraion, a storage agent is installed on the client system. Tivoli Storage Manager supports both tape libraries and FILE libraries. This feature will support SCSI, 349X, and ACSLS libraries.

The configuration procedure you follow will depend on the type of environment you implement, however in all cases you must do the following:

1. Install and configure the client.
2. Install and configure the storage agent.
3. Configure Tivoli Storage Manager client policy for LAN-free data movement.
4. Configure the libraries for LAN-free data movment.
5. Define the libraries and associated paths.
6. Define associated devices and their paths.
7. Configure the server for use with the storage agent.

For more information on configuring Tivoli Storage Manager for LAN-free data movement, including specific installation instructions, see *TSM Managed System for SAN Storage Agent User's Guide*.

Configuring Tivoli Storage Manager for use with Tivoli Data Protection for NDMP

The capabilities of Tivoli Storage Manager are extended through a separately licensed product called Tivoli Data Protection. Tivoli Data Protection allows Tivoli Storage Manager to use Network Data Management Protocol (NDMP) to communicate with NAS file servers and provide backup and restore services. This feature supports SCSI libraries.

In order to implement Tivoli Data Protection for NDMP, you must do the following:

1. Define the libraries and their associated paths.
2. Define a device class for NDMP operations.
3. Define the storage pool for NDMP media.
4. Configure Tivoli Storage Manager policy for NDMP operations.
5. Register the NAS nodes with the server.
6. Define a data mover for the NAS file server.
7. Define the drives and their associated paths.

For more information on configuring Tivoli Storage Manager for use with Tivoli Data Protection for NDMP, including specific installation instructions, see "Chapter 6. Setting Up Tivoli Data Protection for NDMP" on page 97.

Defining Devices and Paths

The following sections describe how to define libraries and drives, as well as their paths, to Tivoli Storage Manager. See "Managing Libraries" on page 131, "Managing Drives" on page 133, and "Managing Paths" on page 138for information about displaying library, drive, and path information, and updating and deleting libraries and drives.

Defining Libraries

Task	Required Privilege Class
Define or update libraries	System or unrestricted storage

Before you can use a drive, you must first define the library to which the drive belongs. This is true for both manually mounted drives and drives in automated libraries. For example, you have several stand-alone tape drives. You could define a library named MANUALMOUNT for these drives by using the following command:

```
define library manualmount libtype=manual
```

For all libraries other than manual libraries, you must also define a path from the server once you have defined the library. For more information about paths, see "Defining Paths" on page 95. For example, if you have an Exabyte EXB-120 device, you could define a library named ROBOTMOUNT using the following command:

```
define library robotmount libtype=scsi
```

Next, you use the DEFINE PATH command. In the path, you must specify the DEVICE parameter. The DEVICE parameter is required and specifies the device special file by which the library's robotic mechanism is known.

```
define path server1 robotmount srctype=server desttype=library
device=/dev/lb0
```

If you have an IBM 3494 Tape Library Dataserver, you can define a library named AUTOMOUNT using the following command:

```
define library automount libtype=349x
```

Next, assuming that you have defined one LMCP whose device name is `/dev/lmcp0`, you define a path for the library:

```
define path server1 automount srctype=server desttype=library
device=/dev/lmcp0
```

Defining Drives

Task	Required Privilege Class
Define drives	System or unrestricted storage

To inform the server about a drive that can be used to access storage volumes, issue the DEFINE DRIVE command, followed by the DEFINE PATH command. For more information about paths, see "Defining Paths" on page 95. When issuing the DEFINE DRIVE command, you must provide some or all of the following information:

Library name
The name of the library in which the drive resides.

Drive name
The name assigned to the drive.

Element address
The element address of the drive. The ELEMENT parameter applies only to SCSI libraries. The element address is a number that indicates the physical location of a drive within an automated library. The server needs the element address to connect the physical location of the drive to the drive's SCSI address. You can get the element address from this Web site: http://www.tivoli.com/support/storage_mgr/tivolimain.html.

For example, to define a drive that belongs to the manual library named MANLIB, enter this command:

```
define drive manlib tapedrv3
```

Next, you would define the path from the server to the drive using the device name used to access the drive:

```
define path server1 tapedrv3 srctype=server desttype=drive library=manlib
device=/dev/mt3
```

Defining Data Movers

Data movers are SAN-attached devices that, through a request from Tivoli Storage Manager, transfer client data for backup, archiving or restore purposes. Data movers are defined as unique objects to Tivoli Storage Manager. Types of data mover devices include NAS file servers.

When issuing the DEFINE DATAMOVER command, you must provide some or all of the following information:

Data Mover Name
The name of the defined data mover.

Type The type of data mover being defined.

High Level Address
The High Level Address is either the numerical IP address or the domain name of a NAS file server.

Low Level Address
The Low Level Address specifies the TCP port number used to access a NAS file server.

User ID
The User ID specifies the user ID for a user when intiating a Network Data Management Protocol (NDMP) session with a NAS file server.

Password
The password specifies the password for a user ID in order to log onto a NAS file server.

Online
The online parameter specifies whether the data mover is online or not.

Data Format
The data format parameter specifes the data format used according to the type of data mover device used.

For example, to define a NAS data mover named NAS1, you would enter the following:

```
define datamover nas1 type=nas hladdress=netapp2.tucson.ibm.com
lladdress=10000 userid=root password=admin dataformat=netappdump
```

Defining Paths

Before a device can be used, a path must be defined between the device and the server or the device and the data mover responsible for outboard data movement. This command must be used to define the following path relationships:

- Between a server and a drive or library.
- Between a data mover and drive or library.

When issuing the DEFINE PATH command, you must provide some or all of the following information:

Source Name
The name of the server or data mover being defined as the source for the path.

Destination Name
The assigned name of the device being defined as the destination for the path.

Source Type
The type of source being defined for the path.

Destination Type
The type of device being defined as the destination for the path.

Library Name
The name of the library that a drive is defined to if the drive is the destination of the path.

Device
The special file name of the device being defined. This parameter may be used when defining a path between a server or a NAS data mover and a library or drive.

For example, if you have a SCSI type library named AUTO8MMLIB that has a device name of /dev/lb3, you could define the path to the server named ASTRO1 by doing the following:

```
define path astro1 auto8mmlib srctype=server desttype=library
device=/dev/lb3
```

If you had a drive, DRIVE01, that resided in library AUTO8MMLIB, and had a device name of /dev/mt4, and you wanted to define it to server ASTRO1, you would do the following:

```
define path astro1 drive01 srctype=server desttype=drive library=auto8mmlib
device=/dev/mt4
```

Chapter 6. Setting Up Tivoli Data Protection for NDMP

This chapter is about planning, configuring, and managing a backup environment that protects your network-attached storage (NAS) file server by using Tivoli Data Protection for NDMP. Tivoli Data Protection for NDMP is a separately licensed product that allows Tivoli Storage Manager to use Network Data Management Protocol (NDMP) to support NAS file servers.

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Requirements

You must meet the following requirements when setting up Tivoli Data Protection for NDMP:

Tivoli Data Protection for NDMP
A separately licensed program product.

NAS File Server
A NAS file server such as Network Appliance File Server. The operating system on the file server must be supported by Tivoli Storage Manager. Visit http://www.tivoli.com/support/storage_mgr/tivolimain.html.

The combination of file server model and operating system must be supported by the NAS file server. Visit http://netapp.com/products/filer/index.html for details.

© Copyright IBM Corp. 1993, 2002

Tape Library

A SCSI-attached or Fibre Channel-attached library that is supported by the Tivoli Storage Manager device driver. Visit http://www.tivoli.com/support/storage_mgr/tivolimain.html.

The tape library is controlled by the Tivoli Storage Manager server using one of the following methods:

- The Tivoli Storage Manager server directly controls the library, which is attached to the Tivoli Storage Manager server.
- The Tivoli Storage Manager server indirectly controls the library. The library is attached to the NAS file server. The Tivoli Storage Manager server passes SCSI commands to the library through the NAS file server.

Library Sharing: The Tivoli Storage Manager server that performs NDMP operations can be a library manager, but cannot be a library client. If the Tivoli Storage Manager server that performs NDMP operations is a library manager, that server must control the library directly and not by passing commands through the NAS file server.

Tape Drives

One or more tape drives in the tape library. The NAS file server must be able to access the drives. The drives must be supported for tape backup operations by the NAS file server and its operating system. Visit http://netapp.com/products/filer/index.htmlfor details.

Drive Sharing: The tape drives can be shared by the Tivoli Storage Manager server and one or more NAS file servers. Also, if the Tivoli Storage Manager server controls the library directly, the drives can be shared:

- By one or more NAS file servers and one or more Tivoli Storage Manager library clients, if the drives are accessible to the library clients.
- By one or more NAS file servers and one or more Tivoli Storage Manager storage agents, if the drives are accessible to the Tivoli Storage Manager server.

Verify the compatibility of specific combinations of a NAS file server, tape devices, and SAN devices with the hardware manufacturers.

Interfaces Used for Tivoli Data Protection for NDMP

You can use any of the interfaces described in this section to perform NDMP operations. You can schedule an NDMP operation using the BACKUP NODE and RESTORE NODE commands and scheduling the operation as an administrative schedule.

Client Interfaces:

- Backup-archive command-line client (on a Windows NT, Windows 2000, AIX, or 32-bit Sun Solaris system)
- Web client

Server Interfaces:

- Server console
- Command line on the administrative client and the server console

Note: All examples in this chapter use server commands.

- Web administrative interface

The Tivoli Storage Manager Web client interface, available with the backup-archive client, lets you display the file systems of the NAS file server in a graphical view. The client function is not required, but you can use the client interfaces for NDMP operations. Tivoli Storage Manager prompts you for an administrator ID and password when you perform NDMP functions using either of the client interfaces. See *Backup-Archive Installation and User's Guide* for more information about installing and activating client interfaces.

Data Formats for Tivoli Data Protection for NDMP

During backup operations involving NDMP, the NAS filer controls the format of the data written to the library. The NDMP format is not the same as the data format used for traditional Tivoli Storage Manager backups. When you define a NAS file server as a data mover and define a storage pool for NDMP operations, you specify the data format. For example, you would specify NETAPPDUMP if the NAS file server is a Network Appliance device.

Additional data formats will be added as Tivoli Storage Manager adds support for NAS file servers from other vendors.

Planning for Tivoli Data Protection for NDMP

Most of the planning required to implement a Tivoli Data Protection for NDMP environment is related to device configuration. You have choices about how to connect and use the libraries and drives.

Planning for Tape Libraries and Drives used in NDMP Operations

Many of the configuration choices you have for libraries and drives are determined by the hardware features of your libraries. You can set up NDMP operations with any supported library and drives. However, the more features your library has, the more flexibility you can exercise in your implementation.

You might start by answering the following questions:

- Do you want to attach tape library robotics to the Tivoli Storage Manager server or to the NAS file server?
- How do you want to use the tape drives in the library?
  - Dedicate all tape drives to NDMP operations.
  - Dedicate some tape drives to NDMP operations and others to traditional Tivoli Storage Manager operations.
  - Share tape drives between NDMP operations and traditional Tivoli Storage Manager operations.

Determining Where to Attach the Tape Library Robotics

One of the first steps in planning for Tivoli Data Protection for NDMP is to determine where to attach the tape library. You must determine whether to attach the library robotics to the Tivoli Storage Manager server or to the NAS file server. Regardless of where you connect library robotics, tape drives must always be connected to the NAS file server for NDMP operations.

Distance and your available hardware connections are factors to consider. If the library does not have separate ports for robotics control and drive access, the library must be attached to the NAS file server because the NAS file server must have access to the drives. If your library has separate ports for robotics control and drive access, you can choose to attach the library robotics to either the Tivoli Storage Manager server or the NAS file server.

If the NAS file server is at a different location than the Tivoli Storage Manager server, the distance may mean that you must attach the library to the NAS file server.

You have the option of dedicating the library to NDMP operations, or of using the library for NDMP operations as well as most traditional Tivoli Storage Manager operations.

Configuration 1 — Library Connected to the Tivoli Storage Manager Server

In this configuration, the tape library must have separate ports for robotics control and for drive access. In addition, the library must be within Fibre-Channel range or SCSI bus range of both the Tivoli Storage Manager server and the NAS file server.

In this configuration, the Tivoli Storage Manager server controls the library through a direct, physical connection to the library robotics control port. For NDMP operations, the drives in the library are connected directly to the NAS file server. The NAS file server transfers data to the tape drive at the request of the Tivoli Storage Manager server. To also use the drives for Tivoli Storage Manager operations, connect the Tivoli Storage Manager server to the tape drives and define the paths from the Tivoli Storage Manager server to the tape drives. This configuration also supports a Tivoli Storage Manager storage agent having access to the drives for its LAN-free operations, and the Tivoli Storage Manager server can be a library manager.

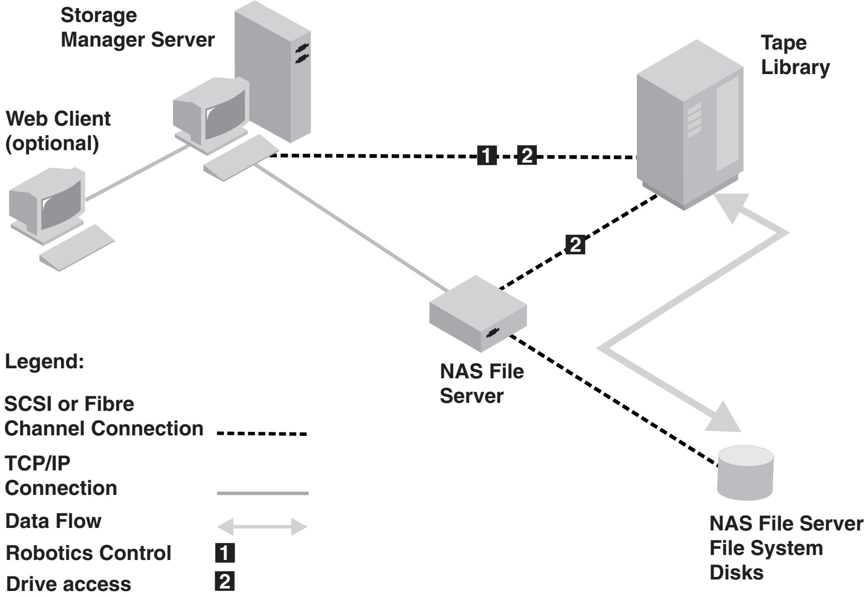


Figure 10. Configuration 1–Library Connected to Tivoli Storage Manager Server

Configuration 2 — Library Connected to the NAS File Server

In this configuration, the library robotics and the drives must be physically connected directly to the NAS file server. No physical connection is required between the Tivoli Storage Manager server and the library.

The Tivoli Storage Manager server controls library robotics by sending library commands across the network to the NAS file server. The NAS file server passes the commands to the tape library. Any responses generated by the library are sent to the NAS file server, and passed back across the network to the Tivoli Storage Manager server. This configuration supports a physically distant Tivoli Storage Manager server and NAS file server. For example, the Tivoli Storage Manager server could be in one city, while the NAS file server and tape library are in another city.

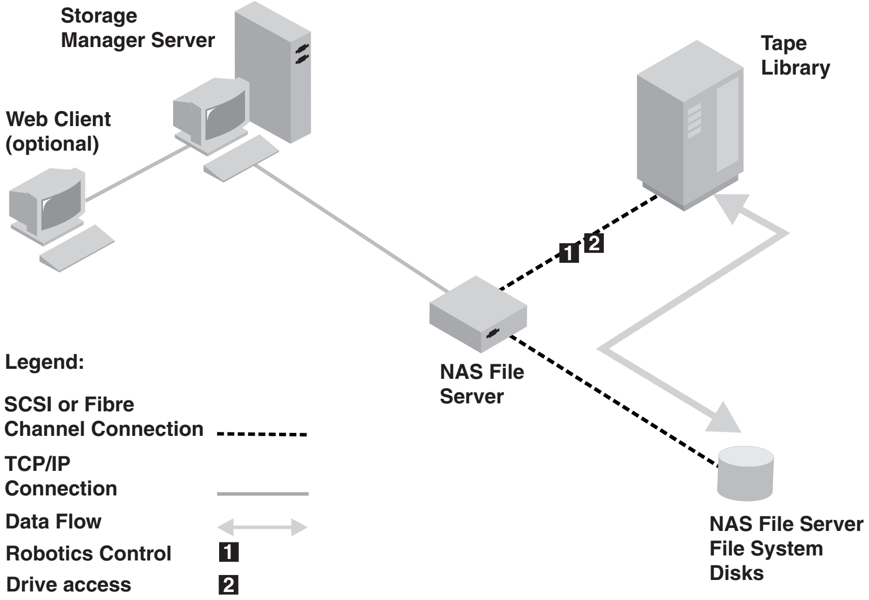


Figure 11. Configuration 2–Library Connected to the NAS File Server

Determining How to Use the Drives in the Library

Drives can be used for multiple purposes because of the flexible configurations allowed by Tivoli Storage Manager. For NDMP operations, the NAS file server must have access to the drive. The Tivoli Storage Manager server can also have access to the same drive, depending on your hardware connections and limitations. All drives are defined to the Tivoli Storage Manager server. However, the same drive may be defined for both traditional Tivoli Storage Manager operations and NDMP operations. Figure 12 on page 103 illustrates one possible configuration. The Tivoli Storage Manager server has access to drives 2 and 3, and each NAS file server has access to drives 1 and 2.

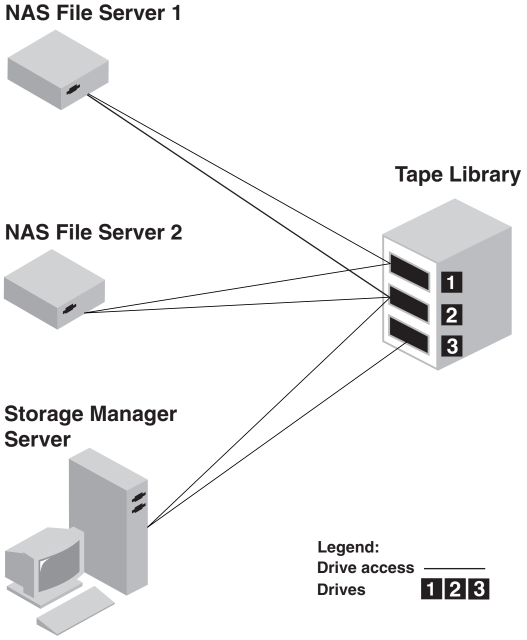


Figure 12. Tivoli Storage Manager Drive Usage Example

To create the configuration shown in the figure, you would do the following:

1. Define all three drives to Tivoli Storage Manager.
2. Define paths from the Tivoli Storage Manager server to drives 2 and 3. Because drive 1 is not accessed by the server, no path is defined.
3. Define each NAS file server as a separate data mover.
4. Define paths from each data mover to drive 1 and to drive 2.

See "Step 6. Defining Tape Drives and Paths for NDMP Operations" on page 107 for more information.

Configuring Tivoli Data Protection for NDMP

Before beginning the configuration of Tivoli Data Protection for NDMP, ensure that you register the license. See "Licensing Tivoli Storage Manager" on page 349.

The following is a checklist to use for configuring Tivoli Data Protection for NDMP:

1. Set up the tape library and media. See "Step 1. Setting Up Tape Libraries for NDMP Operations" on page 104, where the following steps are described in more detail.
   a. Attach the tape library to the NAS file server or to the Tivoli Storage Manager server.
   b. Define the library with a library type of SCSI.
   c. Define a device class for the tape drives.
   d. Define a storage pool for NAS backup media.

2. Configure Tivoli Storage Manager policy for managing NAS image backups. See "Step 2. Configuring Tivoli Storage Manager Policy for NDMP Operations" on page 105.
3. Register a NAS file server node with the Tivoli Storage Manager server. See "Step 3. Registering NAS Nodes with the Tivoli Storage Manager Server" on page 106.
4. Define a data mover for the NAS file server. See "Step 4. Defining a Data Mover for the NAS File Server" on page 106.
5. Define a path from either the Tivoli Storage Manager server or the NAS file server to the library. See "Step 5. Defining a Path to a Library" on page 107.
6. Define the tape drives to Tivoli Storage Manager, and define the paths to those drives from the NAS file server and optionally from the Tivoli Storage Manager server. See "Step 6. Defining Tape Drives and Paths for NDMP Operations" on page 107.
7. Check tapes into the library and label them. See "Step 7. Labeling Tapes and Checking Tapes into the Library" on page 108.
8. Set up scheduled backups for NAS file servers. This step is optional. See "Step 8. Scheduling Tivoli Data Protection for NDMP Operations" on page 108.

Step 1. Setting Up Tape Libraries for NDMP Operations

Before setting up a tape library for NDMP operations, you should have already determined whether you want to attach your library robotics control to the Tivoli Storage Manager server or to the NAS file server. See "Planning for Tape Libraries and Drives used in NDMP Operations" on page 99.

A. Connect the Library and Drives for NDMP Operations

Connect the tape library robotics to the Tivoli Storage Manager server or to the NAS file server. See the manufacturer's documentation for instructions.

Library Connected to Tivoli Storage Manager

Make a SCSI or Fibre Channel connection between the Tivoli Storage Manager server and the library robotics control port. Then connect the NAS file server with the drives you want to use for NDMP operations.

Library Connected to NAS File Server

Make a SCSI or Fibre Channel connection between the NAS file server and the library robotics and drives.

B. Define the Library for NDMP Operations

Define the tape library to Tivoli Storage Manager. For example:

```
define library tsmlib libtype=scsi
```

C. Define a Device Class for NDMP Operations

Create a device class for NDMP operations. A device class defined with a device type of NAS is not explicitly associated with a specific drive type (for example, 3570 or 8mm). However, we recommend that you define separate device classes for different drive types.

In the device class definition:

- Specify NAS as the value for the DEVTYPE parameter.
- Specify 0 as the value for the MOUNTRETENTION parameter. MOUNTRETENTION=0 is required for NDMP operations.
- Specify a value for the ESTCAPACITY parameter.

For example, to define a device class named NASCLASS for a library named NASLIB and media whose estimated capacity is 40GB, enter the following command:

```
define devclass nasclass devtype=nas library=naslib mountretention=0
 estcapacity=40g
```

D. Define a Storage Pool for NDMP Media

The storage pools you define for NDMP media are different from storage pools you define for Tivoli Storage Manager media because they are defined with different data formats. Tivoli Storage Manager operations use storage pools defined with a NATIVE or NONBLOCK data format. NDMP operations require storage pools with a data format that matches the NAS file server and the backup method to be used. For example, to define a storage pool named NASPOOL for a Network Appliance file server, enter the following command:

```
define stgpool naspool nasclass maxscratch=10 dataformat=netappdump
```

Attention: Ensure that you do not accidentally use storage pools that have been defined for NDMP operations in traditional Tivoli Storage Manager operations. Be especially careful when assigning the storage pool name as the value for the DESTINATION parameter of the DEFINE COPYGROUP command. Unless the destination is a storage pool with the appropriate data format, the backup will fail.

Step 2. Configuring Tivoli Storage Manager Policy for NDMP Operations

Policy allows you to manage the number and retention time of NDMP image backup versions. See “Configuring Policy for Tivoli Data Protection for NDMP” on page 306.

1. Create a policy domain for NAS file servers. For example, to define a policy domain that is named NASDOMAIN, enter the following command:

   ```
   define domain nasdomain description='Policy domain for NAS file servers'
   ```
2. Create a policy set in that domain. For example, to define a policy set named STANDARD in the policy domain named NASDOMAIN, enter the following command:

   ```
   define policyset nasdomain standard
   ```
3. Define a management class, and then assign the management class as the default for the policy set. For example, to define a management class named MC1 in the STANDARD policy set, and assign it as the default, enter the following commands:

   ```
   define mgmtclass nasdomain standard mc1
   ```

   ```
   assign defmgmtclass nasdomain standard mc1
   ```
4. Define a backup copy group in the default management class. The destination must be the storage pool you created for NDMP images. In addition, you can specify the number of backup versions to retain. For example, to define a backup copy group for the MC1 management class where up to four versions of each file system are retained in the storage pool named NASPOOL, enter the following command:

   ```
   define copygroup nasdomain standard mc1 destination=naspool verexists=4
   ```

   Attention: Ensure that management classes used for NDMP operations contain the name of a storage pool defined for NDMP operations. Accidentally mixing storage pools used in NDMP operations and those used in Tivoli Storage Manager operations will cause backups to fail.

5. Activate the policy set. For example, to activate the STANDARD policy set in the NASDOMAIN policy domain, enter the following command:

   ```
   activate policyset nasdomain standard
   ```

   The policy is ready to be used. Nodes are associated with Tivoli Storage Manager policy when they are registered. For more information, see “Step 3. Registering NAS Nodes with the Tivoli Storage Manager Server”.

Applying Policy to Backups Initiated with the Client Interface

When a client node initiates a backup, the policy is affected by the option file for that client node. You can control the management classes that are applied to NDMP backups regardless of which node initiates the backup. You can do this by creating a set of options to be used by the client nodes. The option set can include an `include.fs.nas` statement to specify the management class for NAS file server backups. See “Creating Client Option Sets from the Server” on page 253 for more information.

Step 3. Registering NAS Nodes with the Tivoli Storage Manager Server

Register the NAS file server as a Tivoli Storage Manager node. This node name is used to track the image backups for the NAS file server. For example, to register a NAS file server as a node named NASNODE1, with a password of NASPWD1, in a policy domain named NASDOMAIN, enter the following command:

```
register node nasnode1 naspwd1 domain=nasdomain type=nas
```

If you are using a client option set, specify the option set when you register the node.

You can verify that this node is registered by issuing the following command:

```
query node type=nas
```

When you specify TYPE=NAS only NAS nodes are displayed.

Step 4. Defining a Data Mover for the NAS File Server

Define a data mover for each NAS file server in your Tivoli Data Protection for NDMP environment. The data mover name must match the node name that you specified when you registered the NAS node to the Tivoli Storage Manager server. For example, to define a data mover for a NAS node named NASNODE1, enter the following command:

```
define datamover nasnode1 type=nas hladdress=netapp2 lladdress=10000 userid=root
password=admin dataformat=netappdump
```

In this command:

- The high-level address is an IP address for the NAS file server, either a numerical address or a host name.
- The low-level address is the IP port for Network Data Management Protocol (NDMP) sessions with the NAS file server. The default is port number 10000.
- The user ID is the ID defined to the NAS file server that authorizes an NDMP session with the NAS file server (for this example, the user ID is the administrative ID for the Network Appliance file server).
- The password parameter is a valid password for authentication to an NDMP session with the NAS file server.

- The data format is NETAPPDUMP. This is the data format that the Network Appliance file server uses for tape backup.

Step 5. Defining a Path to a Library

Define a path to the library from either the Tivoli Storage Manager or the NAS file server.

Library Connected to Tivoli Storage Manager

For example, issue the following command to define a path from the server, named SERVER1, to the library named TSMLIB:

```
define path server1 tsmlib srctype=server desttype=library device=/dev/lb1
```

Library Connected to NAS File Server

For example, issue the following command to define a path between a NAS data mover named NASNODE1 and a library named NASLIB, use the following command:

```
define path nasnode1 naslib srctype=datamover desttype=library device=mc0
```

The value of the DEVICE parameter is the special file name for the tape library as it is known to the NAS file server. See “Obtaining Special File Names for Path Definitions”.

Obtaining Special File Names for Path Definitions

When you are creating paths, you must provide special file names for tape libraries and drives. For paths from a NAS data mover, the value of the DEVICE parameter in the DEFINE PATH command is the name by which the NAS file server knows a library or drive. You can obtain these names, known as special file names, by querying the NAS file server. For information about how to obtain names for devices that are connected to a NAS file server, consult the product information for the file server.

For example, for a Network Appliance file server, connect to the file server using telnet and issue the SYSCONFIG command. To display the device names for tape libraries, use this command:

```
sysconfig -m
```

To display the device names for tape drives, use this command:

```
sysconfig -t
```

Step 6. Defining Tape Drives and Paths for NDMP Operations

Define the tape drives that you want to use in NDMP operations and the paths to those drives. Depending on your hardware and network connections, you can use the drives for only NDMP operations, or for both traditional Tivoli Storage Manager operations and NDMP operations. For example,

1. Define a drive named NASDRIVE1 for the library named NASLIB.

   ```
   define drive naslib nasdrive1 element=117
   ```

   Note: When you define SCSI drives to Tivoli Storage Manager, you must specify element numbers for the drives when the library contains more than one drive. Element numbers are available from device manufacturers. Element numbers for tape drives are also available in the device support information available on the Tivoli Web site at http://www.tivoli.com/support/storage_mgr/tivolimain.html.

2. Define a path for the drive:

- For example, if the drive is to be used only for NDMP operations, issue the following command:

  Note: For a drive connected only to the NAS file server, do not specify ASNEEDED for the CLEANFREQUENCY parameter of the DEFINE DRIVE command.
- For example, if a drive is to be used for both Tivoli Storage Manager and NDMP operations, enter the following commands:

  ```
  define path server1 nasdrive1 srctype=server desttype=drive
   library=naslib device=/dev/rmt0
  ```

  ```
  define path nasnode1 nasdrive1 srctype=datamover desttype=drive
   library=naslib device=rst01
  ```

Step 7. Labeling Tapes and Checking Tapes into the Library

You must label the tapes and check the tapes into the tape library. These tasks are the same as for other libraries. See “Labeling Removable Media Volumes” on page 114 for more information.

Step 8. Scheduling Tivoli Data Protection for NDMP Operations

You can schedule the backup or restore of NDMP images by using administrative schedules that process the BACKUP NODE or RESTORE NODE administrative commands. The BACKUP NODE and RESTORE NODE commands can be used only for nodes of TYPE=NAS. See “Backing Up and Restoring NAS File Servers Using Tivoli Data Protection for NDMP” for information about the commands.

For example, to create an administrative schedule called NASSCHED to back up all file systems for a node named NASNODE1, enter the following:

```
define schedule nassched type=administrative cmd='backup node nasnode1' active=yes
starttime=20:00 period=1 perunits=days
```

The schedule is active, and is set to run at 8:00 p.m. every day. See “Chapter 17. Automating Server Operations” on page 367 for more information.

Backing Up and Restoring NAS File Servers Using Tivoli Data Protection for NDMP

After you have completed the steps in “Configuring Tivoli Data Protection for NDMP” on page 103, you are ready for NDMP operations. Use either a client interface or an administrative interface described in “Interfaces Used for Tivoli Data Protection for NDMP” on page 98 to perform a file system image backup. For example, to use the Windows NT backup-archive client interface to back up a file system named /vol/vol1 on a NAS file server named NAS1, enter the following command:

```
dsmc backup nas -nasnodename=nas1 {/vol/vol1}
```

For more information on the command, see *Tivoli Storage Manager for Windows Backup-Archive Installation and User’s Guide* or *Tivoli Storage Manager for UNIX Backup-Archive Installation and User’s Guide*.

Note: Whenever you use the client interface, you are asked to authenticate yourself as a Tivoli Storage Manager administrator before the operation can begin. The administrator ID must have at least client owner authority for the NAS node.

You can perform the same backup operation with a server interface. For example, from the administrative command-line client, back up the file system named /vol/vol1 on a NAS file server named NAS1, by entering the following command:

```
backup node nas1 /vol/vol1
```

You can restore the image using either interface. Backups are identical whether they are backed up by using a client interface or a server interface. For example, suppose you want to restore the image backed up in the previous examples. For this example the file system named /vol/vol1 is being restored to /vol/vol2. Restore the file system with the following command, issued from a Windows backup-archive client interface:

```
dsmc restore nas -nasnodename=nas1 {/vol/vol1} {/vol/vol2}
```

You can choose to restore the file system by using a server interface. For example, to restore the file system name /vol/vol1 to file system /vol/vol2, for a NAS file server named NAS1, enter the following command:

```
restore node nas1 /vol/vol1 /vol/vol2
```

Managing Tivoli Data Protection for NDMP

Administrator activities for Tivoli Data Protection for NDMP include managing:

- NAS nodes
- Data movers
- Tape libraries and drives
- Paths
- Device classes
- Storage pools

Managing NAS File Server Nodes

You can update, query, rename, and remove NAS nodes. For example, assume you have created a new policy domain named NASDOMAIN for NAS nodes and you want to update a NAS node named NASNODE1 to include it in the new domain. You might first query the node.

```
query node nasnode1 type=nas
```

Then you might change the domain of the node with the following command:

```
update node nasnode1 domain=nasdomain
```

Renaming a NAS Node

To rename a NAS node, you must also rename the corresponding NAS data mover; they must both must have the same name. For example, to rename NASNODE1 to NAS1 you must perform the following steps:

1. Delete all paths between data mover NASNODE1 and libraries and between data mover NASNODE1 and drives. See “Deleting Paths” on page 138.
2. Delete the data mover defined for the NAS node. See “Managing Data Movers Used in NDMP Operations” on page 110.
3. To rename NASNODE1 to NAS1, issue the following command:

   ```
   rename node nasnode1 nas1
   ```
4. Define the data mover using the new node name. In this example, you must define a new data mover named NAS1 with the same parameters used to define NASNODE1. See “Step 4. Defining a Data Mover for the NAS File Server” on page 106.

Attention: When defining a new data mover for a node that you have renamed, ensure that the data mover name matches the new node name and that the new data mover parameters are duplicates of the original data mover parameters. Any mismatch between a node name and a data mover name or between new data mover parameters and original data mover parameters can prevent you from establishing a session with the NAS file server.

5. Define a path between the NAS data mover and a library only if the tape library is physically connected directly to the NAS file server. See "Step 5. Defining a Path to a Library" on page 107.
6. Define paths between the NAS data mover and any drives used for NDMP operations. See "Step 6. Defining Tape Drives and Paths for NDMP Operations" on page 107.

Deleting a NAS Node

To delete the NAS node, first delete any file spaces for the node. Then delete any paths from the data mover before deleting the data mover. Then you can enter the following command:

```
remove node nas1
```

Managing Data Movers Used in NDMP Operations

You can update, query, and delete the data movers you define for NAS file servers. For example, if you shut down a NAS file server for maintenance, you might want to take the data mover offline. To do this, first query your data movers to identify the data mover for the NAS file server you want to maintain.

```
query datamover nasnode1
```

Then issue the following command to make the data mover offline:

```
update datamover nasnode1 online=no
```

To delete the data mover, you must first delete any path definitions in which the data mover has been used as the source. Then issue the following command to delete the data mover:

```
delete datamover nasnode1
```

Attention: If the data mover has a path to the library, and you delete the data mover or make the data mover offline, you disable access to the library.

Dedicating a Tivoli Storage Manager Drive to NDMP Operations

If you are already using a drive for Tivoli Storage Manager operations, you can dedicate that drive to NDMP operations. Remove Tivoli Storage Manager server access by deleting the path definition with the following command:

```
delete path server1 nasdrive1 srctype=server desttype=drive library=naslib
```

Managing Storage Pools Containing Tivoli Data Protection for NDMP Media

Managing storage pools for Tivoli Data Protection for NDMP backups is different from managing storage pools containing media for traditional Tivoli Storage Manager backups. You can query and update Tivoli Data Protection for NDMP storage pools. You cannot update the DATAFORMAT parameter.

You cannot back up a storage pool that is used for NDMP backups.

The following DEFINE STGPOOL and UPDATE STGPOOL parameters are ignored for Tivoli Data Protection for NDMP storage pools because storage pool hierarchies, reclamation, and migration are not supported for these storage pools:

MAXSIZE
NEXTSTGPOOL
LOWMIG
HIGHMIG
MIGDELAY
MIGCONTINUE
RECLAIMSTGPOOL
OVFLOLOCATION

Attention: Ensure that you do not accidentally use storage pools that have been defined for NDMP operations in traditional Tivoli Storage Manager operations. Be especially careful when assigning the storage pool name as the value for the DESTINATION parameter of the DEFINE COPYGROUP command. Unless the destination is a storage pool with the appropriate data format, the backup will fail.

Chapter 7. Managing Removable Media Operations

This chapter describes routine removable media operations including the following:

- Preparing media for use (checking volumes into automated libraries and labeling volumes)
- Controlling how and when media are reused
- Ensuring that sufficient media are available
- Responding to TSM requests to operators
- Managing libraries, drives (including drive cleaning), paths, and data movers

See the following sections:

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Preparing Removable Media

When TSM accesses a removable media volume, it checks the volume name in the label header to ensure that the correct volume is accessed. To prepare a volume for use, do the following:

1. Label the volume. Any tape or optical volumes must be labeled before the server can use them. See “Labeling Removable Media Volumes” on page 114.
2. For automated libraries, check the volume into the library. See “Checking New Volumes into a Library” on page 117.

© Copyright IBM Corp. 1993, 2002

Tip: When you use the LABEL LIBVOLUME command with drives in an automated library, you can label and check in the volumes with one command.

3. If the storage pool cannot contain scratch volumes (MAXSCRATCH=0), identify the volume to TSM by name so that it can be accessed later. For details, see "Defining Storage Pool Volumes" on page 167.

   If the storage pool can contain scratch volumes (MAXSCRATCH is set to a non-zero value), skip this step.

Labeling Removable Media Volumes

You can use the LABEL LIBVOLUME command from the server console or an administrative client to check in and label volumes in one operation. When you use the command, you can provide parameters that specify:

- The name of the library where the storage volume is located
- The name of the storage volume
- Whether to overwrite a label on the volume
- Whether to search an automated library for volumes for labeling
- Whether to read media labels:
  - To prompt for volume names in SCSI libraries
  - To read the bar-code label for each cartridge in SCSI, 349X, and ACSLS libraries
- Whether to check in the volume:
  - To add the volume to the scratch pool
  - To designate the volume as private
- The type of device (applies to 349X libraries only)

To use the LABEL LIBVOLUME command, there must be a drive that is not in use by another TSM process. This includes volumes that are mounted but idle. If necessary, use the DISMOUNT VOLUME command to dismount the idle volume to make that drive available.

By default, the LABEL LIBVOLUME command does not overwrite an existing label. However, if you want to overwrite an existing label, you can specify OVERWRITE=YES parameter.

Attention: By overwriting a volume label, you destroy all of the data that resides on the volume. Use caution when overwriting volume labels to avoid destroying important data.

When you use the LABEL LIBVOLUME command, you can identify the volumes to be labeled in one of the following ways:

- Explicitly name one volume.
- Enter a range of volumes by using the VOLRANGE parameter.
- Use the VOLLIST parameter to specify a file that contains a list of volume names or to explicitly name one or more volumes.

For automated libraries, you are prompted to insert the volume in the entry/exit slot of the library. If no I/O convenience station is available, insert the volume in an empty slot. For manual libraries, you are prompted to load the volume directly into a drive.

Labeling Volumes In a Manual Drive

Suppose that you want to label a few new volumes by using a manual tape drive that is defined as */dev/mt5*. The drive is attached at SCSI address 5. Enter the following command:

```
label libvolume tsmlibname volname
```

Note: The LABEL LIBVOLUME command selects the next free drive. If you have more than one free drive, this may not be */dev/mt5*.

If the server is not available, use the following command:

```
> dsmlabel -drive=/dev/mt5
```

The DSMLABEL utility, which is an offline utility for labeling sequential access volumes for TSM, must read the *dsmserv.opt* file to pick up the language option. Therefore, you must issue the DSMLABEL command from the */usr/tivoli/tsm/server/bin/* directory, or you must set the DSMSERV_DIR and DSMSERV_CONFIG environment variables.

Labeling Volumes in a SCSI or ACSLS Library

You can label volumes one at a time or let TSM search the library for volumes.

Labeling Volumes One at a Time: If you choose to label volumes one at a time, do the following:

1. Insert volumes into the library when prompted to do so. The library mounts each inserted volume into a drive.
2. For a SCSI library, enter a volume name when you are prompted (LABELSOURCE=PROMPT). A label is written to the volume using the name that you entered.
3. If the library does not have an entry/exit port, you are prompted to remove the tape from a specified slot number (not a drive). If the library has an entry/exit port, the command by default returns each labeled volume to the entry/exit port of the library.

Labeling New Volumes in a Library: Suppose you want to label a few new volumes in a SCSI library. You want to manually insert each new volume into the library, and you want the volumes to be placed in storage slots inside the library after their labels are written. You know that none of the new volumes contains valid data, so it is acceptable to overwrite existing volume labels. You only want to use one of the library's four drives for these operations.

Note: This example works for libraries that do not have entry and exit ports.
Enter the following command:

```
label libvolume tsmlibname volname overwrite=yes checkin=scratch
```

If the server is not available, use the following command:

```
> dsmlabel -drive=/dev/mt0,116 -library=/dev/lb0 -overwrite -keep
```

Searching the Library: The LABEL LIBVOLUME command searches all of the storage slots in the library for volumes and tries to label each one that it finds. You choose this mode when you specify the SEARCH=YES parameter. After a volume is labeled, the volume is returned to its original location in the library. Specify SEARCH=BULK if you want the server to search the library's entry/exit ports for usable volumes to be labeled.

When you specify LABELSOURCE=PROMPT, the volume is moved from its location in the library or in the entry/exit ports to the drive. The server prompts you to issue the REPLY command containing the label string, and that label is written to the tape.

If the library has a bar-code reader, the LABEL LIBVOLUME command can use the reader to obtain volume names, instead of prompting you for volume names. Use the SEARCH=YES and LABELSOURCE=BARCODE parameters.If you specify the LABELSOURCE=BARCODE parameter, the volume bar code is read, and the tape is moved from its location in the library or in the entry/exit ports to a drive where the bar-code label is written. After the tape is labeled, it is moved back to its location in the library, to the entry/exit ports, or to a storage slot if the CHECKIN option is specified. For bar-code support to work correctly for libraries supported by Tivoli Storage Manager, the Tivoli Storage Manager server, and the device driver must be at the same level. Bar-code support is available for libraries supported by Tivoli Storage Manager and which use the Tivoli Storage Manager device driver or the RMSS Magstar or LTO Ultrium device driver.

Suppose that you want to label all volumes in a SCSI library. The library contains four drives, but you want to use only two of them to label volumes. The drives are at element addresses 116 and 117. Enter the following command:

```
label libvolume tsmlibname search=yes labelsource=barcode
```

Tivoli Storage Manager will select the next available drive.

Note: The LABELSOURCE=BARCODE parameter is valid only for SCSI libraries.

If the server is not available, use the following command:

```
> dsmlabel -drive=/dev/mt0,116 -drive=/dev/mt1,117 -library=/dev/lb0 -search
```

Labeling Volumes in a 349X Library

For a 349X library, the server attempts to label only those volumes in the INSERT category and the library’s private and scratch categories. All other volumes are ignored by the labeling process. This precaution prevents the inadvertent destruction of that data on volumes being actively used by other systems connected to the library device.

Note: The LABEL LIBVOLUME command labels volumes in the INSERT category and in the PRIVATE, 3490SCRATCH, and 3590SCRATCH categories, but not the volumes already checked into the library.

Suppose that you want to label all of the volumes that are in the INSERT category in an IBM 3494 tape library. Enter the following command:

```
label libvolume tsmlibname search=yes devtype=3590
```

Note: If the volumes to be labeled are 3590 media and there are both 3490 and 3590 drives in the library, you must add DEVTYPE=3590.

If the server is not available, use the following command:

```
> dsmlabel -drive=/dev/rmt1 -drive=/dev/rmt2 -library=/dev/lmcp0
```

Labeling Optical Volumes

You can use the LABEL LIBVOLUME command to label optical disks (3.5-inch and 5.25-inch).

```
label libvolume opticlib search=yes labelsource=prompt
```

You can also use the DSMLABEL utility to format and label 3.5-inch and 5.25-inch optical disks. Use the `-format` parameter when starting the DSMLABEL utility.

The DSMLABEL utility, which is an offline utility for labeling sequential access volumes for TSM, must read the *dsmserv.opt* file to pick up the language option. Therefore, you must issue the DSMLABEL command from the */usr/tivoli/tsm/server/bin/* directory, or you must set the DSMSERV_DIR and DSMSERV_CONFIG environment variables.

```
> dsmlabel -drive=/dev/rop1,117 -library=/dev/lb0 -search -format
```

Checking New Volumes into a Library

Task	Required Privilege Class
Inform the server when a new volume is available in an automated library	System or unrestricted storage

To inform the server that a new volume is available in an automated library, check in the volume with the CHECKIN LIBVOLUME or LABEL LIBVOLUME command. When a volume is checked in, the server adds the volume to its library volume inventory. You can use the LABEL LIBVOLUME command to check in and label volumes in one operation.

Note: Do not mix volumes with bar-code labels and volumes without bar-code labels in a library device because bar-code scanning can take a long time for unlabeled volumes.

When you check in a volume, you must supply the name of the library and the status of the volume (private or scratch).

To check in one or just a few volumes, you can specify the name of the volume with the command, and issue the command for each volume. See “Checking Volumes into a SCSI Library One at a Time” on page 118.

To check in a larger number of volumes, you can use the search capability of the CHECKIN command (see “Checking in Volumes in Library Slots” on page 119) or you can use the VOLRANGE parameter of the CHECKIN command.

When using the CHECKIN LIBVOLUME command, be prepared to supply some or all of the following information:

Library name
Specifies the name of the library where the storage volume is to be located.

Volume name
Specifies the volume name of the storage volume being checked in.

Status
Specifies the status that is assigned to the storage volume being checked in. If you check in a volume that has already been defined in a storage pool or in the volume history file, you must specify a volume status of *private* (STATUS=PRIVATE). This status ensures that the volume is not overwritten when a scratch mount is requested. The server does not check in a volume with scratch status when that volume already belongs to a storage pool or is a database, export, or dump volume.

Check label
Specifies whether TSM should read sequential media labels of volumes during CHECKIN command processing, or use a bar-code reader. See “Checking Media Labels” on page 119.

For optical volumes being checked in to an automated library, you must specify CHECKLABEL=YES. TSM must read the label to determine the type of volume: rewritable (OPTICAL device type) or write-once read-many (WORM or WORM12 device type).

Swap Specifies whether TSM will initiate a swap operation when an empty slot is not available during CHECKIN command processing. See "Allowing Swapping of Volumes When the Library Is Full" on page 120.

Mount wait
Specifies the maximum length of time, in minutes, to wait for a storage volume to be mounted.

Search
Specifies whether TSM searches the library for volumes that have not been checked in. See "Checking Volumes into a SCSI Library One at a Time", "Checking in Volumes in Library Slots" on page 119, and"Checking in Volumes in Library Entry/Exit Ports" on page 119.

Device type
This parameter only applies to 349X libraries containing 3590 devices. This parameter allows you to specify the device type for the volume being checked in.

Checking Volumes into a SCSI Library One at a Time

Specify SEARCH=NO if you want to check in only a single volume that is not currently in the library. TSM requests that the mount operator load the volume in the entry/exit port of the library.

If the library does not have an entry/exit port, TSM requests that the mount operator load the volume into a slot within the library. The request specifies the location with an *element address*. For any library or medium changer that does not have an entry/exit port, you need to know the element addresses for the cartridge slots and drives. If there is no worksheet listed for your device in http://www.tivoli.com/support/storage_mgr/tivolimain.html, see the documentation that came with your library.

Note: Element addresses are sometimes numbered starting with a number other than one. Check the worksheet to be sure.

For example, to check in volume VOL001 manually, enter the following command:

```
checkin libvolume tapelib vol001 search=no status=scratch
```

Iif the library has an entry/exit port, you are prompted to insert a cartridge into the entry/exit port. If the library does not have an entry/exit port, you are prompted to insert a cartridge into one of the slots in the library. Element addresses identify these slots. For example, TSM finds that the first empty slot is at element address 5. The message is:

```
ANR8306I 001: Insert 8MM volume VOL001 R/W in slot with element
address 5 of library TAPELIB within 60 minutes; issue 'REPLY' along
with the request ID when ready.
```

Check the worksheet for the device if you do not know the location of element address 5 in the library. See http://www.tivoli.com/support/storage_mgr/tivolimain.html to find the worksheet. When you have inserted the volume as requested, respond to the message from a TSM administrative client. Use the request number (the number at the beginning of the mount request):

```
reply 1
```

Checking Volumes into a 349X Library One at a Time

Specify SEARCH=NO for a 349X library, to search for volumes that have already been inserted into the library via the convenience or bulk I/O station.

```
checkin libvolume 3494lib vol001 search=no status=scratch
```

If the volume has already been inserted, the server finds and processes it. If not, you can insert the volume into the I/O station during the processing of the command.

Checking in Volumes in Library Slots

Specify SEARCH=YES if you want the server to search the library slots for new volumes that have not already been added to the library volume inventory. Use this mode when you have a large number of volumes to check in, and you want to avoid issuing an explicit CHECKIN LIBVOLUME command for each volume. For example, for a SCSI library you can simply open the library access door, place all of the new volumes in unused slots, close the door, and issue the CHECKIN LIBVOLUME command with SEARCH=YES.

If you are using a 349X library, the server searches only for new volumes in the following categories:

- INSERT
- TSM's private category (PRIVATECATEGORY, specified when you define the library)
- TSM's scratch category (SCRATCHCATEGORY, specified when you define the library)

  If 3590 support is enabled, the server searches for two scratch categories: SCRATCHCATEGORY, and SCRATCHCATEGORY + 1.

This restriction prevents the server from using volumes owned by another application that is accessing the library simultaneously.

Checking in Volumes in Library Entry/Exit Ports

Specify SEARCH=BULK if you want TSM to search the library's entry/exit ports for volumes that can be checked in automatically. For SCSI libraries, the server scans all of the entry/exit ports in the library for volumes. If a volume is found that contains a valid volume label, it is checked in automatically. The CHECKLABEL option NO is invalid with this SEARCH option. When you use the CHECKLABEL=YES parameter, the volume is moved from the entry/exit ports to the drive where the label is read. After reading the label, the tape is moved from the drive to a storage slot. When you use the CHECKLABEL=BARCODE parameter, the volume's bar code is read and the tape is moved from the entry/exit port to a storage slot. For bar-code support to work correctly, the TSM or RMSS device driver must be installed for TSM-controlled libraries.

Checking Media Labels

When you check in a volume, you can specify whether TSM should read the labels of the media during check-in processing. When label-checking is on, TSM mounts each volume to read the internal label and only checks in a volume if it is properly labeled. This can prevent future errors when volumes are actually used in storage pools, but also increases processing time at check in. For information on how to label new volumes, see "Preparing Removable Media" on page 113.

If a library has a bar-code reader and the volumes have bar-code labels, you can save time in the check in process. TSM uses the characters on the label as the name for the volume being checked in. If a volume has no bar-code label, TSM mounts the volumes in a drive and attempts to read the recorded label. For example, to use the bar-code reader to check in all volumes found in the TAPELIB library as scratch volumes, enter the following command:

```
checkin libvolume tapelib search=yes status=scratch checklabel=barcode
```

Allowing Swapping of Volumes When the Library Is Full

If no empty slots are available in the library when you are checking in volumes, the check-in fails unless you allow *swapping*. If you allow swapping and the library is full, TSM selects a volume to eject before checking in the volume you requested.

Use the CHECKIN LIBVOLUME command to allow swapping. When you specify YES for the SWAP parameter, TSM initiates a swap operation if an empty slot is not available to check in a volume. TSM ejects the volume that it selects for the swap operation from the library and replaces the ejected volume with the volume that is being checked in. For example:

```
checkin libvolume auto wpdv00 swap=yes
```

TSM selects the volume to eject by checking first for any available scratch volume, then for the least frequently mounted volume.

Managing the Volume Inventory

With TSM, you manage your volume inventory by performing the following tasks:

- Controlling TSM access to volumes
- Reusing tapes in storage pools
- Reusing volumes used for database backups and export operations
- Maintaining a supply of scratch volumes

Controlling Access to Volumes

TSM expects to be able to access all volumes it knows about. For example, TSM tries to fill up tape volumes. If a volume containing client data is only partially full, TSM will later request that volume be mounted to store additional data. If the volume cannot be mounted, an error occurs.

To make volumes that are not full available to be read but not written to, you can change the volume access mode. For example, use the UPDATE VOLUME command with ACCESS=READONLY. The server will not attempt to mount a volume that has an access mode of unavailable.

If you want to make volumes unavailable to send the data they contain offsite for safekeeping, a more controlled way to do this is to use a copy storage pool. You can back up your primary storage pools to a copy storage pool and then send the copy storage pool volumes offsite. You can track these copy storage pool volumes by changing their access mode to offsite, and updating the volume history to identify their location. For more information, see “Backing Up Storage Pools” on page 507.

Reusing Tapes in Storage Pools

To reuse tapes in storage pools, you must do two things:

Expiration Processing of Client Files

Expiration processing deletes from the database information about any client files that are expired (no longer valid according to the policies you have set). For example, suppose four backup versions of a file exist in server storage, and only three versions are allowed in the backup policy (the management class) for the file. Expiration processing deletes information about the oldest of the four versions of the file. The space that the file occupied in the storage pool can then be reclaimed.

You can run expiration processing automatically or by command. See "Running Expiration Processing to Delete Expired Files" on page 302.

Reclamation of Volumes

You can have TSM reclaim volumes that pass a *reclamation threshold*, a percentage of unused space on the volume. TSM moves data to consolidate valid, unexpired files onto fewer tapes. The reclamation threshold is set for each storage pool. See "Reclaiming Space in Sequential Access Storage Pools" on page 189.

For a storage pool associated with a library that has more than one drive, the reclaimed data is moved to other volumes in the same storage pool. For a storage pool associated with a library that has only one drive, the reclaimed data is moved to volumes in another storage pool that you must define, called a reclamation storage pool. See "Reclaiming Volumes in a Storage Pool with One Drive" on page 192.

Setting Up a Tape Rotation

Over time, media ages, and the data on backup may no longer be needed. You can reclaim useful data on media and then reclaim and reuse the media themselves. When you set up expiration processing, you can determine when data is no longer needed. See "File Expiration and Expiration Processing" on page 273.

TSM policy determines how many backup versions are retained and how long they are retained. See "Basic Policy Planning" on page 270.

Deleting Data - Expiration Processing

Expiration processing deletes data that is no longer valid either because it exceeds the retention specifications in policy or because users or administrators have deleted the active versions of the data. See "Running Expiration Processing to Delete Expired Files" on page 302.

Reusing Media - Reclamation Processing

Data on tapes may expire, move, or be deleted. Reclamation processing consolidates any unexpired data by moving it from multiple volumes onto fewer volumes. The media can then be returned to the storage pool and reused.

You can set a reclamation threshold that allows TSM to reclaim volumes whose valid data drops below a threshold. The threshold is a percentage of unused space on the volume and is set for each storage pool. The amount of data on the volume and the reclamation threshold for the storage pool affects when the volume is reclaimed. See "Reclaiming Space in Sequential Access Storage Pools" on page 189.

Determining When Media Have Reached End of Life

You can use TSM to display statistics about volumes including the number of write operations performed on the media and the number of write errors. TSM overwrites this statistical data for media initially defined as scratch volumes each time the media are reclaimed. For media initially defined as private volumes, TSM maintains this statistical data, even as the volume is reclaimed. You can compare the information with the number of write operations and write errors recommended by the manufacturer.

Reclaim any valid data from volumes that have reached end of life. If the volumes are in automated libraries, check them out of the volume inventory. Delete private volumes the database with the DELETE VOLUME command. See "Reclaiming Space in Sequential Access Storage Pools" on page 189.

Ensuring Media are Available for the Tape Rotation

Over time, the demand for volumes may cause the storage pool to run out of space. You can set the maximum number of scratch volumes high enough to meet demand by doing one or both of the following:

- Increase the maximum number of scratch volumes by updating the storage pool definition. Label and check in new volumes to be used as scratch volumes if needed.
- Make volumes available for reuse by running expiration processing and reclamation, to consolidate data onto fewer volumes. See "Reusing Tapes in Storage Pools" on page 120.

For automated libraries, see "Managing Server Requests for Media" on page 127.

Write-once-read-many (WORM) drives can waste media when TSM cancels transactions because volumes are not available to complete the backup. Once TSM writes to WORM volumes, the space on the volumes cannot be reused, even if the transactions are canceled (for example, if a backup is canceled because of a shortage of media in the device).

Large files can cause even greater waste. For example, consider a client backing up a 12GB file onto WORM platters that hold 2.6GB each. If the backup requires five platters and only four platters are available, TSM cancels the backup and the four volumes that were written to cannot be reused.

To minimize wasted WORM media:

1. Ensure that the maximum number of scratch volumes for the device storage pool is at least equal to the number of storage slots in the library.
2. Check enough volumes into the device's volume inventory for the expected load.

If most backups are small files, controlling the transaction size can affect how WORM platters are used. Smaller transactions mean that less space is wasted if a transaction such as a backup must be canceled. Transaction size is controlled by a server option, TXNGROUPMAX, and a client option, TXNBYTELIMIT.

Reusing Volumes Used for Database Backups and Export Operations

When you back up the database or export server information, TSM records information about the volumes used for these operations in the *volume history* file. TSM will not allow you to reuse these volumes until you delete the volume information from the volume history file. To reuse volumes that have previously been used for database backup or export, use the DELETE VOLHISTORY command. For information about the volume history file, see "Saving the Volume History File" on page 514.

Note: If your server is licensed for the DRM product, the volume information is automatically deleted during MOVE DRMEDIA command processing. For additional information about DRM, see “Chapter 23. Using Tivoli Disaster Recovery Manager” on page 545.

Maintaining a Supply of Scratch Volumes

When you define a storage pool, you must specify the maximum number of scratch volumes that the storage pool can use. TSM automatically requests a scratch volume when needed. When the number of scratch volumes that TSM is using for the storage pool exceeds the maximum number of scratch volumes specified, the storage pool can run out of space.

Ensure that you set the maximum number of scratch volumes high enough for the expected usage. When you exceed this number, you can do one or both of the following:

- Increase the maximum number of scratch volumes by updating the storage pool definition. Label new volumes to be used as scratch volumes if needed.
- Make volumes available for reuse by running expiration processing and reclamation, to consolidate data onto fewer volumes. See “Reusing Tapes in Storage Pools” on page 120.

For automated libraries, see also “Maintaining a Supply of Scratch Volumes in an Automated Library” on page 127.

Maintaining a Supply of Volumes in a WORM Library

For libraries with WORM drives, prevent cancellation of data storage transactions by maintaining a supply of scratch or new private volumes in the library. Canceled transactions can cause wasted WORM media. TSM cancels (rolls back) a transaction if volumes, either private or scratch, are not available to complete the data storage operation. After TSM begins a transaction by writing to a WORM volume, the written space on the volume cannot be reused, even if the transaction is canceled.

For example, if a client starts to back up data and does not have sufficient volumes in the library, TSM cancels the backup transaction. The WORM volumes to which TSM had already written for the canceled backup are wasted because the volumes cannot be reused. Suppose that you have WORM platters that hold 2.6GB each. A client starts to back up a 12GB file. If TSM cannot acquire a fifth scratch volume after filling four volumes, TSM cancels the backup operation. The four volumes that TSM already filled cannot be reused.

To minimize cancellation of transactions, do the following:

- Ensure that you have enough volumes available in the library to handle expected client operations such as backup.
  - Verify that you set the maximum number of scratch volumes for the storage pool that is associated with the library to a high enough number.
  - Check enough scratch or private volumes into the library to handle the expected load.
- If your clients tend to store files of smaller sizes, controlling the transaction size can affect how WORM platters are used. Smaller transactions waste less space if a transaction such as a backup must be canceled. The TXNGROUPMAX server option and the TXNBYTELIMIT client option control transaction size. See “How the Server Groups Files before Storing” on page 171 for information.

Managing Volumes in Automated Libraries

TSM tracks the scratch and private volumes available in an automated library through a *library volume inventory*. TSM maintains an inventory for each automated library. The library volume inventory is separate from the inventory of volumes for each storage pool. To add a volume to a library's volume inventory, you *check in* a volume to that TSM library. For details on the check-in procedure, see "Checking New Volumes into a Library" on page 117.

To ensure that TSM's library volume inventory remains accurate, you must *check out* volumes when you need to physically remove volumes from a SCSI, 349X, or ACSLS library. When you check out a volume that is being used by a storage pool, the volume remains in the storage pool. If TSM requires the volume to be mounted while it is checked out, a message to the mount operator's console is displayed with a request to check in the volume. If the check in is not successful, TSM marks the volume as unavailable.

While a volume is in the library volume inventory, you can change its status from scratch to private.

To check whether TSM's library volume inventory is consistent with the volumes that are physically in the library, you can audit the library. The inventory can become inaccurate if volumes are moved in and out of the library without informing the server via volume check-in or check-out.

Task	Required Privilege Class
Changing the status of a volume in an automated library Removing volumes from a library Returning volumes to a library	System or unrestricted storage

Changing the Status of a Volume

The UPDATE LIBVOLUME command lets you change the status of a volume in an automated library from scratch to private, or private to scratch. However, you cannot change the status of a volume from private to scratch if the volume belongs to a storage pool or is defined in the volume history file. You can use this command if you make a mistake when checking in volumes to the library and assign the volumes the wrong status.

Removing Volumes from a Library

You may want to remove a volume from an automated library. The following two examples illustrate this:

- You have exported data to a volume in the library and want to take it to another system for an import operation.
- All of the volumes in the library are full, and you want to remove some that are not likely to be accessed to make room for new volumes that can be used to store more data.

To remove a volume from an automated library, use the CHECKOUT LIBVOLUME command. By default, the server mounts the volume being checked out and verifies the internal label. When the label is verified, the server removes the volume from the library volume inventory, and then moves it to the entry/exit port or convenience I/O station. of the library. If the library does not have an entry/exit port, TSM requests that the mount operator remove the volume from a slot within the library.

For SCSI libraries with multiple entry/exit ports, use the REMOVE=BULK parameter of the CHECKOUT LIBVOLUME command to eject the volume to the next available entry/exit port.

If you check out a volume that is defined in a storage pool, the server may attempt to access it later to read or write data. If this happens, the server requests that the volume be checked in.

Returning Volumes to a Library

When you check out a volume that is defined to a storage pool, to make the volume available again, do the following:

1. Check in the volume for the library, with private status. Use the CHECKIN LIBVOLUME command with the parameter STATUS=PRIVATE.
2. If the volume was marked unavailable, update the volume's ACCESS value to read/write or read-only. Use the UPDATE VOLUME command with the ACCESS parameter.

Managing a Full Library

As TSM fills volumes in a storage pool, the number of volumes needed for the pool may exceed the physical capacity of the library. To make room for new volumes while keeping track of existing volumes, you can define a storage pool overflow location near the library. You then move media to the overflow location as needed. The following shows a typical sequence of steps to manage a full library:

1. Define or update the storage pool associated with the automated library, including the overflow location parameter. For example, you have a storage pool named ARCHIVEPOOL associated with an automated library. Update the storage pool to add an overflow location of Room2948. Enter this command:
   ```
   update stgpool archivepool ovflocation=Room2948
   ```
2. When the library becomes full, move the full volumes out of the library and to the overflow location that you defined for the storage pool. For example, to move all full volumes in the specified storage pool out of the library, enter this command:
   ```
   move media * stgpool=archivepool
   ```

   All full volumes are checked out of the library. TSM records the location of the volumes as Room2948. You can use the DAYS parameter to specify the number of days that must elapse before a volume is eligible for processing by the MOVE MEDIA command.
3. Check in new scratch volumes, if needed.
4. Reuse the empty scratch storage volumes in the overflow location. For example, enter this command:
   ```
   query media * stg=* whereovflocation=Room2948 wherestatus=empty
   move media * stg=* wherestate=mountablenotinlib wherestatus=empty
   cmd="checkin libvol autolib &vol status=scratch"
   cmdfilename=/tsm/move/media/checkin.vols
   ```

   For more information, see *Administrator's Reference*.

5. As requested through TSM mount messages, check in volumes that TSM needs for operations. The mount messages include the overflow location of the volumes.

To find the overflow location of a storage pool, you can use the QUERY MEDIA command. This command can also be used to generate commands. For example, you can issue a QUERY MEDIA command to get a list of all volumes in the overflow location, and at the same time generate the commands to check in all those volumes to the library. For example, enter this command:

```
query media format=cmd stgpool=archivepool whereovflocation=Room2948
cmd="checkin libvol autolib &vol status=private"
cmdfilename="/tsm/move/media/checkin.vols"
```

Use the DAYS parameter to specify the number of days that must elapse before the volumes are eligible for processing by the QUERY MEDIA command.

The file that contains the generated commands can be run using the TSM MACRO command. For this example, the file may look like this:

```
checkin libvol autolib TAPE13 status=private
checkin libvol autolib TAPE19 status=private
```

Auditing a Library's Volume Inventory

Task	Required Privilege Class
Audit the volume inventory of a library	System or unrestricted storage

You can audit an automated library to ensure that the library volume inventory is consistent with the volumes that physically reside in the library. You may want to do this if the library volume inventory is disturbed due to manual movement of volumes in the library or database problems. Use the AUDIT LIBRARY command to restore the inventory to a consistent state. Missing volumes are deleted, and the locations of the moved volumes are updated. However, new volumes are not added during an audit.

Unless your SCSI library has a bar-code reader, the server mounts each volume during the audit to verify the internal labels on volumes. For 349X libraries, the server uses the information from the Library Manager.

Issue the AUDIT LIBRARY command only when there are no volumes mounted in the library drives. If any volumes are mounted but in the IDLE state, you can issue the DISMOUNT VOLUME command to dismount them.

If a SCSI library has a bar-code reader, you can save time by using the bar-code reader to verify the identity of volumes. If a volume has a bar-code label, the server uses the characters on the label as the name for the volume. The volume is not mounted to verify that the bar-code name matches the internal volume name. If a volume has no bar-code label, the server mounts the volume and attempts to read the recorded label. For example, to audit the TAPELIB library using its bar-code reader, issue the following command:

```
audit library tapelib checklabel=barcode
```

Maintaining a Supply of Scratch Volumes in an Automated Library

When you define a storage pool that is associated with an automated library (through the device class), you must specify a maximum number of scratch volumes equal to the physical capacity of the library. When the number of scratch volumes that TSM is using for the storage pool exceeds that number, do the following:

1. Add scratch volumes to the library by checking in volumes. Label them if necessary.

   You may need to use an overflow location to move volumes out of the library to make room for these scratch volumes. See "Maintaining a Supply of Scratch Volumes" on page 123.
2. Increase the maximum number of scratch volumes by updating the storage pool definition. The increase should equal the number of scratch volumes that you checked in.

Managing Server Requests for Media

TSM displays requests and status messages to all administrative clients that are started in console mode. These request messages often have a time limit. If the request is not fulfilled within the time limit, the operation times out and fails.

For manual libraries, TSM detects when there is a cartridge loaded in a drive, and no operator reply is necessary. For automated libraries, commands such as CHECKIN LIBVOLUME, LABEL LIBVOLUME, and CHECKOUT LIBVOLUME involve inserting or removing cartridges from the library and issuing a reply message.

Using the Administrative Client for Mount Messages

The server sends mount request status messages to the server console and to all administrative clients in mount mode or console mode parameter. For example, to start an administrative client in mount mode, enter this command:

```
> dsmadmc -mountmode
```

Mount Operations for Manual Libraries

Volumes are mounted as a result of mount requests from TSM. For manual libraries, you can monitor the mount requests on the server console or through an administrative client in mount mode or console mode. Someone you designate as the operator must respond to the mount requests by putting in tape volumes as requested.

Handling Messages for Automated Libraries

For automated libraries, mount messages are sent to the library and not to an operator. Messages about problems with the library are sent to the mount message queue. You can see these messages on administrative clients in mount mode or console mode. However, you cannot use the TSM REPLY command to respond to these messages.

Requesting Information about Pending Operator Requests

Task	Required Privilege Class
Request information about operator requests or mounted volumes	Any administrator

You can get information about pending operator requests either by using the QUERY REQUEST command or by checking the mount message queue on an administrative client started in mount mode.

When you issue the QUERY REQUEST command, TSM displays requested actions and the amount of time remaining before the requests time out. For example, you enter the command as follows:

```
query request
```

The following shows an example of a response to the command:

```
ANR8352I Requests outstanding:
ANR8326I 001: Mount 8MM volume DSM001 R/W in drive TAPE01 (/dev/mt1)
of MANUAL8MM within 60 minutes.
```

Replying to Operator Requests

Task	Required Privilege Class
Reply to operator requests	Operator

When the server requires that an explicit reply be provided when a mount request is completed, you can reply with the REPLY command. The first parameter for this command is the request identification number that tells the server which of the pending operator requests has been completed. This 3-digit number is always displayed as part of the request message. It can also be obtained by issuing a QUERY REQUEST command. If the request requires the operator to provide a device to be used for the mount, the second parameter for this command is a device name.

For example, enter the following command to respond to request 001 for tape drive TAPE01:

```
reply 1
```

Canceling an Operator Request

Task	Required Privilege Class
Cancel operator requests	Operator

If a mount request for a manual library cannot be satisfied, you can issue the CANCEL REQUEST command. This command forces the server to cancel the request and cause the operation that needed the requested volume to fail.

The CANCEL REQUEST command must include the request identification number. This number is included in the request message. You can also obtain it by issuing a QUERY REQUEST command, as described in "Requesting Information about Pending Operator Requests" on page 128.

You can specify the PERMANENT parameter if you want to mark the requested volume as UNAVAILABLE. This process is useful if, for example, the volume has been moved to a remote site or is otherwise inaccessible. By specifying PERMANENT, you ensure that the server does not try to mount the requested volume again.

For most of the requests associated with automated (SCSI) libraries, an operator must perform a hardware or system action to cancel the requested mount. For such requests, the CANCEL REQUEST command is not accepted by the server.

Responding to Requests for Volume Check-In

If the server cannot find a particular volume to be mounted in an automated library, the server requests that the operator check in the volume. For example, a client requests that an archived file be retrieved. The file was archived in a storage pool in an automated library. The server looks for the volume containing the file in the automated library, but cannot find the volume. The server then requests that the volume be checked in.

If the volume that the server requests is available, put the volume in the library and check in the volume using the normal procedures ("Checking New Volumes into a Library" on page 117).

If the volume requested is unavailable (lost or destroyed), update the access mode of the volume to UNAVAILABLE by using the UPDATE VOLUME command. Then cancel the server's request for check-in by using the CANCEL REQUEST command. (Do *not* cancel the client process that caused the request.) To get the ID of the request to cancel, use the QUERY REQUEST command.

If you do not respond to the server's check-in request within the mount-wait period of the device class for the storage pool, the server marks the volume as unavailable.

Determining Which Volumes Are Mounted

Task	Required Privilege Class
Request information about which volumes are mounted	Operator

For a report of all volumes currently mounted for use by the server, you can issue the QUERY MOUNT command. The report shows which volumes are mounted, which drives have accessed them, and if the volumes are currently being used.

Dismounting an Idle Volume

Task	Required Privilege Class
Request a volume dismount	Operator

After a volume becomes idle, the server keeps it mounted for a time specified by the mount retention parameter for the device class. Use of mount retention can reduce the access time if volumes are repeatedly used.

An administrator can explicitly request to dismount an idle volume by issuing the DISMOUNT VOLUME command. This command causes the server to dismount the named volume from the drive in which it is currently mounted.

For information about setting mount retention times, see "Mount Retention Period" on page 144.

Performing Operations with Shared Libraries

The library client contacts the library manager, when the library manager starts and the storage device initializes, or after a library manager is defined to a library client. The library client confirms that the contacted server is the library manager for the named library device. The library client also compares drive definitions with the library manager for consistency. The library client contacts the library manager for each of the following operations:

Volume Mount

A library client sends a request to the library manager for access to a particular volume in the shared library device. For a scratch volume, the library client does not specify a volume name. If the library manager cannot access the requested volume, or if scratch volumes are not available, the library manager denies the mount request. If the mount is successful, the library manager returns the name of the drive where the volume is mounted.

Volume Release (free to scratch)

When a library client no longer needs to access a volume, it notifies the library manager that the volume should be returned to scratch. The library manager's database is updated with the volume's new location. The volume is deleted from the volume inventory of the library client.

Table 11 shows the interaction between library clients and the library manager in processing Tivoli Storage Manager operations.

Table 11. How SAN-enabled Servers Process Tivoli Storage Manager Operations

Operation (Command)	**Library Manager**	**Library Client**
Query library volumes (QUERY LIBVOLUME)	Displays the volumes that are checked into the library. For private volumes, the owner server is also displayed.	Not applicable.
Check in and check out library volumes (CHECKIN LIBVOLUME, CHECKOUT LIBVOLUME)	Performs the commands to the library device.	Not applicable. When a check-in operation must be performed because of a client restore, a request is sent to the library manager server.

Table 11. How SAN-enabled Servers Process Tivoli Storage Manager Operations (continued)

Operation (Command)	Library Manager	Library Client
Move media and move DRM media (MOVE MEDIA, MOVE DRMEDIA)	Only valid for volumes used by the library manager server.	Requests that the library manager server perform the operations. Generates a checkout process on the library manager server.
Audit library inventory (AUDIT LIBRARY)	Performs the inventory synchronization with the library device.	Performs the inventory synchronization with the library manager server.
Label a library volume (LABEL LIBVOLUME)	Performs the labeling and check-in of media.	Not applicable.
Dismount a volume (DISMOUNT VOLUME)	Sends the request to the library device.	Requests that the library manager server perform the operation.
Query a volume (QUERY VOLUME)	Checks whether the volume is owned by the requesting library client server and checks whether the volume is in the library device.	Requests that the library manager server perform the operation.

Managing Libraries

You can query, update, and delete libraries.

Requesting Information About Libraries

Task	Required Privilege Class
Request information about libraries	Any administrator

You can request information about one or more libraries by using the QUERY LIBRARY command. You can request either a standard or a detailed report. For example, to display information about all libraries, issue the following command:

```
query library
```

The following shows an example of the output from this command.

```
Library    Library   Private     Scratch     External
Name       Type      Category    Category    Manager
-------    -------   --------    --------    --------
MANLIB     MANUAL
EXB        SCSI
3494LIB    349X      300         301
```

Updating Libraries

You can update an existing library by issuing the UPDATE LIBRARY command. To update the device names of a library, issue the UPDATE PATH command.

Note: You cannot update a MANUAL library.

Task	Required Privilege Class
Update libraries	System or unrestricted storage

Automated Libraries

If your system or device is reconfigured, and the device name changes, you may need to update the device name. The examples below show how you can use the UPDATE LIBRARY and UPDATE PATH commands for the following library types:

- SCSI
- 349X
- ACSLS
- External

Examples:

- **SCSI Library**

  Update the path from SERVER1 to a SCSI library named SCSILIB:

  ```
  update path server1 scsilib srctype=server desttype=library device=/dev/lb1
  ```

  Update the definition of a SCSI library named SCSILIB defined to a library client so that a new library manager is specified:

  ```
  update library scsilib primarylibmanager=server2
  ```

- **349X Library**

  Update the path from SERVER1 to an IBM 3494 library named 3494LIB with new device names.

  ```
  update path server1 3494lib srctype=server desttype=library
   device=/dev/lmcp1,/dev/lmcp2,/dev/lmcp3
  ```

  Update the definition of an IBM 3494 library named 3494LIB defined to a library client so that a new library manager is specified:

  ```
  update library 3494lib primarylibmanager=server2
  ```

- **ACSLS Library**

  Update an ACSLS library named ACSLSLIB with a new ID number.

  ```
  update library acslslib ascid=1
  ```

- **External Library**

  Update an external library named EXTLIB with a new media manager path name.

  ```
  update path server1 extlib srctype=server desttype=library
   externalmanager=/v/server/mediamanager.exe
  ```

  Update an EXTERNAL library named EXTLIB in a LAN-free configuration so that the server uses the value set for mount retention in the device class associated with the library:

  ```
  update library extlib obeymountretention=yes
  ```

Deleting Libraries

Task	Required Privilege Class
Delete libraries	System or unrestricted storage

Before you delete a library with the DELETE LIBRARY command, you must delete all of the drives that have been defined as part of the library and delete the path to the library. See “Deleting Drives” on page 138.

For example, suppose that you want to delete a library named 8MMLIB1. After deleting all of the drives defined as part of this library and the path to the library, issue the following command to delete the library itself:

```
delete library 8mmlib1
```

Managing Drives

You can query, update, and delete drives.

Requesting Information about Drives

Task	**Required Privilege Class**
Request information about drives	Any administrator

You can request information about drives by using the QUERY DRIVE command. This command accepts wildcard characters for both a library name and a drive name. See *Administrator’s Reference* for information about this command and the use of wildcard characters.

For example, to query all drives associated with your server, enter the following command:

```
query drive
```

The following shows an example of the output from this command.

```
Library   Drive    Device     On Line
Name      Name     Type
--------  -------  ---------  -------
MANLIB    8MM.0    8MM        Yes
AUTOLIB   8MM.2    8MM        Yes
```

Updating Drives

You can change the attributes of a drive by issuing the UPDATE DRIVE command.

Task	**Required Privilege Class**
Update drives	System or unrestricted storage

You can change the following attributes of a drive by issuing the UPDATE DRIVE command.

- The element address, if the drive resides in a SCSI library
- The ID of a drive in an ACSLS library
- The cleaning frequency
- Change whether the drive is online or offline

For example, to change the element address of a drive named DRIVE3 to 119, issue the following command:

```
update drive auto drive3 element=119
```

If you are reconfiguring your system, you can change the device name of a drive by issuing the UPDATE PATH command. For example, to change the device name of a drive named DRIVE3, issue the following command:

```
update path server1 drive3 srctype=server desttype=drive library=scsilib device=/dev/rmt0
```

Note: You cannot change the element number or the device name if a drive is in use. See "Taking Drives Offline". If a drive has a volume mounted, but the volume is idle, it can be explicitly dismounted. See "Dismounting an Idle Volume" on page 129.

Taking Drives Offline

You can take a drive offline while it is in use. For example, you might take a drive offline for another activity, such as maintenance. If you take a drive offline while it is in use, the mounted volume completes its current process. If this volume was part of a series of volumes in a transaction, the drive is no longer available to complete mounting the series. If no other drives are available, the active process may fail. The offline state is retained even if the server is halted and brought up again. If a drive is marked offline when the server is brought up, a warning is issued noting that the drive must be manually brought online. If all the drives in a library are taken offline, processes requiring a library mount point will fail, rather than queue up for one.

The ONLINE parameter specifies the value of the drive's online state, even if the drive is in use. ONLINE=YES indicates that the drive is available for use. ONLINE=NO indicates that the drive is not available for use (offline). Do not specify other optional parameters along with the ONLINE parameter. If you do, the drive will not be updated, and the command will fail when the drive is in use. You can specify the ONLINE parameter when the drive is involved in an active process or session, but this is not recommended.

Cleaning Drives

Task	Required Privilege Class
Clean drives	System or unrestricted storage

The server can control cleaning tape drives in SCSI libraries and offers partial support for cleaning tape drives in manual libraries. For automated library devices, you can automate cleaning by specifying the frequency of cleaning operations and checking a cleaner cartridge into the library's volume inventory. TSM mounts the cleaner cartridge as specified. For manual library devices, TSM issues a mount request for the cleaner cartridge.

Deciding Whether the Server Controls Drive Cleaning

Some SCSI libraries such as STK 9710 and IBM 3575 provide automatic drive cleaning. In such cases, choose either the library drive cleaning or the Tivoli Storage Manager drive cleaning, but not both. Manufacturers that include library cleaning recommend its use to prevent premature wear on the read/write heads of the drives.

Drives and libraries from different manufacturers differ in how they handle cleaner cartridges and how they report the presence of a cleaner cartridge in a drive. The device driver may not be able to open a drive that contains a cleaner cartridge. Sense codes and error codes that are issued by devices for drive cleaning vary. Library drive cleaning is usually transparent to all applications. Therefore, Tivoli

Storage Manager may not always detect cleaner cartridges in drives and may not be able to determine when cleaning has begun.

Some devices require a small amount of idle time between mount requests to start drive cleaning. However, Tivoli Storage Manager tries to minimize the idle time for a drive. The result may be to prevent the library drive cleaning from functioning effectively. If this happens, try using Tivoli Storage Manager to control drive cleaning. Set the frequency to match the cleaning recommendations from the manufacturer.

If you have Tivoli Storage Manager control drive cleaning, disable the library drive cleaning function to prevent problems. If the library drive cleaning function is enabled, some devices automatically move any cleaner cartridge found in the library to slots in the library that are dedicated for cleaner cartridges. An application does not know that these dedicated slots exist. You will not be able to check a cleaner cartridge into the Tivoli Storage Manager library inventory until you disable the library drive cleaning function.

Cleaning Drives in an Automated Library

Set up server-controlled drive cleaning in an automated library with these steps:

1. Define or update the drives in a library, using the CLEANFREQUENCY parameter. The CLEANFREQUENCY parameter sets how often you want the drive cleaned. Refer to the DEFINE DRIVE and UPDATE DRIVE commands. Consult the manuals that accompany the drives for recommendations on cleaning frequency.

   For example, to have DRIVE1 cleaned after 100GB is processed on the drive, issue the following command:

   ```
   update drive autolib1 drive1 cleanfrequency=100
   ```

   Consult the drive manufacturer's information for cleaning recommendations. If the information gives recommendations for cleaning frequency in terms of hours of use, convert to a gigabytes value by doing the following:

   a. Use the bytes-per-second rating for the drive to determine a gigabytes-per-hour value.

   b. Multiply the gigabytes-per-hour value by the recommended hours of use between cleanings.

   c. Use the result as the cleaning frequency value.

   Note: For IBM 3570 and 3590 drives, we recommend that you specify a value for the CLEANFREQUENCY parameter rather than specify ASNEEDED. Using the cleaning frequency recommended by the product documentation will not overclean the drives.

2. Check a cleaner cartridge into the library's volume inventory with the CHECKIN LIBVOLUME command. For example:

   ```
   checkin libvolume autolib1 cleanv status=cleaner cleanings=10 checklabel=no
   ```

   After the cleaner cartridge is checked in, the server will mount the cleaner cartridge in a drive when the drive needs cleaning. The server will use that cleaner cartridge for the number of cleanings specified. See "Checking In Cleaner Cartridges" on page 136 and "Operations with Cleaner Cartridges in a Library" on page 136 for more information.

For details on the commands, see *Administrator's Reference*.

Checking In Cleaner Cartridges: You must check a cleaner cartridge into an automated library's volume inventory to have the server control drive cleaning without further operator intervention.

It is recommended that you check in cleaner cartridges one at a time and do not use the search function of check-in for a cleaner cartridge.

Attention: When checking in a cleaner cartridge to a library, ensure that it is correctly identified to the server as a cleaner cartridge. Also use caution when a cleaner cartridge is already checked in and you are checking in data cartridges. Ensure that cleaner cartridges are in their correct home slots, or errors and delays can result.

When checking in data cartridges with SEARCH=YES, ensure that a cleaner cartridge is not in a slot that will be detected by the search process. Errors and delays of 15 minutes or more can result from a cleaner cartridge being improperly moved or placed. For best results, check in the data cartridges first when you use the search function. Then check in the cleaner cartridge separately.

For example, if you need to check in both data cartridges and cleaner cartridges, put the data cartridges in the library and check them in first. You can use the search function of the CHECKIN LIBVOLUME command (or the LABEL LIBVOLUME command if you are labeling and checking in volumes). Then check in the cleaner cartridge to the library by using one of the following methods.

- Check in without using search:

  ```
  checkin libvolume autolib1 cleanv status=cleaner cleanings=10
   checklabel=no
  ```

  The server then requests that the cartridge be placed in the entry/exit port, or into a specific slot.
- Check in using search, but limit the search by using the VOLRANGE or VOLLIST parameter:

  ```
  checkin libvolume autolib1 status=cleaner cleanings=10 search=yes checklabel=barcode
   vollist=cleanv
  ```

  The process scans the library by using the bar-code reader, looking for the CLEANV volume.

Manual Drive Cleaning in an Automated Library: If your library has limited capacity and you do not want to use a slot in your library for a cleaner cartridge, you can still make use of the server's drive cleaning function. Set the cleaning frequency for the drives in the library. When a drive needs cleaning based on the frequency setting, the server issues the message, ANR8914I. For example:

```
ANR89141I Drive DRIVE1 in library AUTOLIB1 needs to be cleaned.
```

You can use that message as a cue to manually insert a cleaner cartridge into the drive. However, the server cannot track whether the drive has been cleaned.

Operations with Cleaner Cartridges in a Library: When a drive needs to be cleaned, the server runs the cleaning operation after dismounting a data volume if a cleaner cartridge is checked in to the library. If the cleaning operation fails or is cancelled, or if no cleaner cartridge is available, then the indication that the drive needs cleaning is lost. Monitor cleaning messages for these problems to ensure that drives are cleaned as needed. If necessary, use the CLEAN DRIVE command to have the server try the cleaning again, or manually load a cleaner cartridge into the drive.

The server uses a cleaner cartridge for the number of cleanings that you specify when you check in the cleaner cartridge. If you check in more than one cleaner cartridge, the server uses one of them for its designated number of cleanings. Then the server begins to use the next cleaner cartridge.

Visually verify that cleaner cartridges are in the correct storage slots before issuing any of the following commands:

- AUDIT LIBRARY
- CHECKIN LIBVOLUME with SEARCH specified
- LABEL LIBVOLUME with SEARCH specified

To find the correct slot for a cleaner cartridge, use the QUERY LIBVOLUME command.

Cleaning Drives in a Manual Library

Cleaning a drive in a manual library is the same as setting up drive cleaning without checking in a cleaner cartridge for an automated library. The server issues the ANR8914I message when a drive needs cleaning. For example:

```
ANR89141I Drive DRIVE1 in library MANLIB1 needs to be cleaned.
```

Monitor the activity log or the server console for these messages and load a cleaner cartridge into the drive as needed. The server cannot track whether the drive has been cleaned.

Error Checking for Drive Cleaning

Occasionally an administrator might move some cartridges around within a library and put a data cartridge where TSM shows that there is a cleaner cartridge. TSM uses the process in this section to recover from the error. When a drive needs cleaning, the server loads what its database shows as a cleaner cartridge into the drive. The drive then moves to a READY state, and TSM detects that the cartridge is a data cartridge. The server then performs the following steps:

1. The server attempts to read the internal tape label of the data cartridge.
2. The server ejects the cartridge from the drive and moves it back to the home slot of the “cleaner” cartridge within the library. If the eject fails, the server marks the drive offline and issues a message that the cartridge is still in the drive.
3. The server checks out the “cleaner” cartridge to avoid selecting it for another drive cleaning request. The “cleaner” cartridge remains in the library but no longer appears in the TSM library inventory.
4. If the server was able to read the internal tape label, the server checks the volume name against the current library inventory, storage pool volumes, and the volume history file.
   - If there is not a match, an administrator probably checked in a data cartridge as a cleaner cartridge by mistake. Now that the volume is checked out, you do not need to do anything else.
   - If there is a match, the server issues messages that manual intervention and a library audit are required. Library audits can take considerable time, so an administrator should issue the command when sufficient time permits. See “Auditing a Library’s Volume Inventory” on page 126.

Deleting Drives

You can delete a drive by issuing the DELETE DRIVE command.

Task	Required Privilege Class
Delete drives	System or unrestricted storage

A drive cannot be deleted if it is currently in use. If a drive has a volume mounted, but the volume is currently idle, it can be dismounted as described in "Dismounting an Idle Volume" on page 129. A drive cannot be deleted until the defined path to the drive has been deleted. Also, a library cannot be deleted until all of the drives defined within it are deleted.

Managing Paths

You can query, update, and delete paths.

Requesting Information About Paths

By using the QUERY PATH command, you can obtain information about paths. You can request either a standard or a detailed report. This command accepts wildcard characters for both a source name and a destination name. See *Administrator's Reference* for information about this command and the use of wildcard characters.

For example, to display information about all paths, issue the following command:

```
query path
```

The following shows an example of the output from this command.

```
Source Name  Source Type  Destination Name  Destination Type  Online
-----------  -----------  ----------------  ----------------  ------
SERVER1      server       TSMLIB            Library           Yes
NETAPP1      Data mover   DRIVE1            Drive             Yes
NETAPP1      Data mover   NASLIB            Library           Yes
datamover2   Data mover   drive4            Drive             Yes
```

Updating Paths

You can update an existing path by issuing the UPDATE PATH command. The examples below show how you can use the UPDATE PATH commands for the following path types:

- **Library Paths**

  Update the path to change the device name for a SCSI library named SCSILIB:

  ```
  update path server1 scsilib srctype=server desttype=library device=/dev/lb1
  ```

- **Drive Paths**

  Update the path to change the device name for a drive named NASDRV1:

  ```
  update path nas1 nasdrv1 srctype=datamover desttype=drive
   library=naslib device=/dev/mt1
  ```

Deleting Paths

Task	Required Privilege Class
Delete paths	System or unrestricted storage

A path cannot be deleted if the destination is currently in use. Before you can delete a path to a device, you must delete the device.

Delete a path from a NAS data mover NAS1 to the library NASLIB.

```
delete path nas1 naslib srctype=datamover desttype=library
```

Attention: If you delete the path to a device or make the path offline, you disable access to that device.

Managing Data Movers

You can query, update, and delete data movers.

Requesting Information About Data Movers

By using the QUERY DATAMOVER command, you can obtain information about SCSI and NAS data movers. You can request either a standard or a detailed report. For example, to display a standard report about all data movers, issue the following command:

```
query datamover *
```

The following shows an example of the output from this command.

```
Data Mover Name    Type      Online
-------------    ----------  -------
NASMOVER1           NAS         Yes
NASMOVER2           NAS          No
DATAMOVER1         SCSI         Yes
```

Updating Data Movers

You can update an existing data mover by issuing the UPDATE DATAMOVER command.

For example, to update the data mover for the node named NAS1 to change the IP address, issue the following command:

```
update datamover nas1 hladdress=9.67.97.109
```

Deleting Data Movers

Before you can delete a data mover, you must delete all paths defined for the data mover.

To delete a data mover named NAS1, issue the following command:

```
delete datamover nas1
```

Chapter 8. Defining Device Classes

A device class represents a device type that can be used by Tivoli Storage Manager. The server uses device class definitions to determine which types of devices and volumes to use to:

- Store backup, archive, or space-managed data in primary storage pools
- Store copies of primary storage pool data in copy storage pools
- Store database backups
- Export or import TSM data

One device class can be associated with multiple storage pools, but each storage pool is associated with only one device class.

For random access storage, TSM supports only the DISK device class, which is defined by TSM. However, you can define many storage pools associated with the DISK device class.

See the following sections:

Tasks:
"Defining and Updating Device Classes for Tape Devices" on page 143
"Defining and Updating Device Classes for Generic Tape Devices" on page 146
"Defining and Updating Device Classes for Optical Devices" on page 146
"Defining and Updating Device Classes for Removable File System Devices" on page 147
"Defining and Updating FILE Device Classes" on page 148
"Defining and Updating SERVER Device Classes" on page 149
"Requesting Information about a Device Class" on page 151
"How TSM Fills Volumes" on page 152
"Deleting a Device Class" on page 151

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Defining and Updating Device Classes for Sequential Media

Task	**Required Privilege Class**
Define or update device classes	System or unrestricted storage

© Copyright IBM Corp. 1993, 2002

For sequential access storage, TSM supports the following device types:

Device Type	Media Type	Device Examples
3570	IBM 3570 cartridges	IBM 3570 drives
3590	IBM 3590 cartridges	IBM 3590, 3590E drives
4MM	4mm cartridges	IBM 7206-005
8MM	8mm cartridges	IBM 7208-001 and 7208-011
CARTRIDGE	Tape cartridges	IBM 3480, 3490, and 3490E drives
DLT	Digital linear tape (DLT) cartridges	DLT2000, DLT4000, DLT7000 and DLT8000 drives
DTF	Digital tape format (DTF) cartridges	Sony GY-2120, Sony DMS-8400 drives
ECARTRIDGE	Tape cartridges	StorageTek SD-3 and 9490 drives
FILE	File system or storage volumes	Server
GENERICTAPE	Tape cartridges	Tape drives supported by operating system device drivers
LTO	LTO Ultrium cartridges	IBM 3580, 3581, 3583, 3584
NAS	Unknown	Tape drives supported by the NAS file server for backups
OPTICAL	5.25-inch rewritable optical cartridges	5.25-inch optical drives
QIC	Quarter-inch tape cartridges	IBM 7207
REMOVABLEFILE	Iomega Zip or Jaz drives, or CDROM media	Removable media devices that are attached as local, removable file systems
SERVER	Storage volumes or files archived in another TSM server	TSM target server
WORM	5.25-inch write-once read-many (WORM) optical cartridges	5.25-inch optical drives
WORM12	12-inch write-once ready-many optical cartridges	12–inch optical drives
WORM14	14-inch write-once ready-many optical cartridges	14–inch optical drives

You can define multiple device classes for each device type. For example, you may need to specify different attributes for different storage pools that use the same type of tape drive. Variations may be required that are not specific to the device, but rather to how you want to use the device (for example, mount retention or mount limit).

For all device types other than FILE or SERVER, you must define libraries and drives to TSM before you define the device classes.

If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command. When you use this option, the files specified are automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

The following sections explain the device classes for each supported device type.

Defining and Updating Device Classes for Tape Devices

To use tape devices, you must define a device class by issuing a DEFINE DEVCLASS command with the DEVTYPE parameter.

Note: For upgrades from TSM Enhanced Version 2 to Version 3 or later, you must define a device class with the DEVTYPE parameter set to DLT or ECART.

Other parameters specify how to manage data storage operations involving the new device class:
- MOUNTLIMIT
- MOUNTWAIT
- MOUNTRETENTION
- PREFIX
- FORMAT
- ESTCAPACITY
- LIBRARY

You can update the device class by issuing the UPDATE DEVCLASS command.

Mount Limit

The MOUNTLIMIT parameter specifies the maximum number of volumes that can be simultaneously mounted for a device class. You can limit the number of drives that the device class has access to at one time with the MOUNTLIMIT parameter.

The default mount limit value is DRIVES. The DRIVES parameter indicates that every time a mount point is allocated, the number of drives online and defined to the library is used to calculate the true mount limit value. The maximum value for this parameter is 256 and the minimum value is 0. A zero value prevents new transactions from gaining access to the storage pool.

When selecting a mount limit for a device class, be sure to consider the following questions:

- How many storage devices are connected to your system?

  Do not specify a mount limit value that is greater than the number of associated available drives in your installation. If the server tries to mount as many volumes as specified by the mount limit and no drives are available for the required volume, an error occurs and client sessions may be terminated. (This does not apply when the DRIVES parameter is specified.)

  Note: TSM cannot share drives between multiple device classes.
- Are you using the simultaneous write function to primary and copy storage pools?

  Specify a mount limit value that provides a sufficient number of mount points to support a simultaneous write to the primary storage pool and all associated copy storage pools.
- Are you associating multiple device classes of the same device type with a single library?

A device class associated with a library can use any drive in the library that is compatible with the device class' device type. Because you can associate more than one device class with a library, a single drive in the library can be used by more than one device class. However, TSM does not manage how a drive is shared among multiple device classes.

When you associate multiple device classes of the same device type with a library, add up the mount limits for all these device classes. Ensure that this sum is no greater than the number of compatible drives.

- How many TSM processes do you want to run at the same time, using devices in this device class?

  TSM automatically cancels some processes to run other, higher priority processes. If the server is using all available drives in a device class to complete higher priority processes, lower priority processes must wait until a drive becomes available. For example, TSM cancels the process for a client backing up directly to tape if the drive being used is needed for a server migration or tape reclamation process. TSM cancels a tape reclamation process if the drive being used is needed for a client restore operation. For additional information, see "Preemption of Client or Server Operations" on page 361.

  If processes are often canceled by other processes, consider whether you can make more drives available for TSM use. Otherwise, review your scheduling of operations to reduce the contention for drives.

  This consideration also applies to the primary and copy storage pool simultaneous write function. You must have enough drives available to allow for a successful simultaneous write.

Note: If the library associated with this device class is EXTERNAL type, it is recommended that you explicitly specify the mount limit instead of using MOUNTLIMIT=DRIVES.

Mount Wait Period

The MOUNTWAIT parameter specifies the maximum amount of time, in minutes, that the server waits for a drive to become available for the current mount request. The default mount wait period is 60 minutes. The maximum value for this parameter is 9999 minutes.

Note: This parameter is not valid for EXTERNAL library types.

Mount Retention Period

The MOUNTRETENTION parameter specifies the amount of time that a mounted volume should remain mounted after its last I/O activity. If this idle time limit is reached, the server dismounts the volume. The default mount retention period is 60 minutes. The maximum value for this parameter is 9999 minutes.

Note: A device class with DEVType=NAS allows only a value of zero (0).

For example, if the mount retention value is 60, and a mounted volume remains idle for 60 minutes, then the server dismounts the volume.

If a volume is used frequently, you can improve performance by setting a longer mount retention period to avoid unnecessary mount and dismount operations.

If mount operations are being handled by manual, operator-assisted activities, you may want to use a large mount retention period. For example, if only one operator supports your entire operation on a weekend, then define a long mount retention period so that the operator is not being asked to mount volumes every few minutes.

While TSM has a volume mounted, the drive is allocated to TSM and cannot be used for anything else. If you need to free the drive for other uses, you can cancel TSM operations that are using the drive and then dismount the volume. For example, you can cancel server migration or backup operations. For information on how to cancel processes and dismount volumes, see “Canceling Server Processes” on page 361 and “Dismounting an Idle Volume” on page 129.

Tape Label Prefix

By using the PREFIX parameter, you can specify a prefix value that is used to construct the *file name* string that is stored in the label area of each tape volume.

The prefix string is used as the prefix of the file name that is written to the label of each tape. The default value for the tape label prefix string is ADSM.

Note: This parameter is used primarily in the OS/390 and z/OS operating systems.

Recording Format

You can use the FORMAT parameter to specify the recording format used by TSM when writing data to removable media. See the *Administrator's Reference* for information about the recording formats for each device type.

Specify FORMAT=DRIVE parameter only if all drives associated with that device class are identical. If some drives associated with the device class support a higher density format than others and you specify FORMAT=DRIVE, mount errors can occur. For example, suppose a device class uses two incompatible devices such as an IBM 7208-2 and an IBM 7208-12. The server might select the high-density recording format of 8500 for each of two new volumes. Later, if the two volumes are to be mounted concurrently, one fails because only one of the drives is capable of the high-density recording format.

The recording format that TSM uses for a given volume is selected when the first piece of data is written to the volume. Updating the FORMAT parameter does not affect media that already contain data until those media are rewritten from the beginning. This process may happen after a volume is reclaimed or deleted, or after all of the data on the volume expires.

Estimated Capacity

The ESTCAPACITY parameter specifies the estimated capacity for volumes assigned to this device class. TSM estimates the capacity of the volumes in a storage pool based on the parameters assigned to the device class associated with the storage pool. For tape device classes, the default values selected by the server depend on the recording format used to write data to the volume. You can either accept the default for a given device type or specify a value.

Note: For a device class with DEVTYPE=NAS, this value is required.

See *Administrator's Reference* for information about the estimated capacities of recording formats for each device type.

TSM also uses estimated capacity to determine when to begin reclamation storage pool volumes. For more information on how TSM uses the estimated capacity value, see “How TSM Fills Volumes” on page 152.

Library

Before the server can mount a volume, it must know which drives can be used to satisfy the mount request. This process is done by specifying the library when the device class is defined. The library must contain drives that can be used to mount the volume.

Only one library can be associated with a given device class. However, multiple device classes can reference the same library. Unless you are using the DRIVES value for MOUNTLIMIT, you must ensure that the numeric value of the mount limits of all device classes do not exceed the number of drives defined in the referenced library.

There is no default value for this parameter. It is required, and so must be specified when the device class is defined.

Defining and Updating Device Classes for Generic Tape Devices

To use a tape device that is supported by an operating system device driver, you must define a device class whose device type is GENERICTAPE.

For a manual library with multiple drives of device type GENERICTAPE, ensure that the device types and recording formats of the drives are compatible. Because the devices are controlled by the operating system device driver, the TSM server is not aware of the following:

- The actual type of device: 4mm, 8mm, digital linear tape, and so forth. For example, if you have a 4mm device and an 8mm device, you must define separate manual libraries for each device.
- The actual cartridge recording format. For example, if you have a manual library defined with two device classes of GENERICTAPE, ensure the recording formats are the same for both drives.

You can update the device class information by issuing the UPDATE DEVCLASS command. Other parameters, in addition to device type, specify how to manage server storage operations:

Mount Limit
See “Mount Limit” on page 143.

Mount Wait Period
See “Mount Wait Period” on page 144.

Mount Retention Period
See “Mount Retention Period” on page 144.

Estimated Capacity
You can specify an estimated capacity value of any volumes defined to a storage pool categorized by a GENERICTAPE device class. The default ESTCAPACITY value for a volume in a GENERICTAPE device class is 1GB. Specify a capacity appropriate for your particular tape drive.

Library
See “Library”.

Defining and Updating Device Classes for Optical Devices

To use optical media, you must define a device class by issuing the DEFINE DEVCLASS command with a DEVTYPE parameter for one of the optical devices:

Parameter	Description
OPTICAL	5.25-inch rewritable optical media
WORM	5.25-inch write-once optical media
WORM12	12-inch write-once optical media.
WORM14	14-inch write once optical media.

Other parameters specify how to manage data storage operations involving the new device class:

Mount Limit
See "Mount Limit" on page 143.

Mount Wait Period
See "Mount Wait Period" on page 144.

Mount Retention
See "Mount Retention Period" on page 144.

Recording Format
See "Recording Format" on page 145.

Estimated Capacity
See "Estimated Capacity" on page 145.

Library
See "Library" on page 146.

You can update the device class information by issuing the UPDATE DEVCLASS command.

Defining and Updating Device Classes for Removable File System Devices

Removable file devices include devices such as Iomega Zip drives or Jaz drives and CD-ROM drives. Define a device class for these devices by issuing the DEFINE DEVCLASS command with the DEVTYPE=REMOVABLEFILE parameter. See "Configuring Removable File Devices" on page 86 for more information.

Other parameters specify how to manage storage operations involving the new device class:

Mount Wait
See "Mount Wait Period" on page 144.

Mount Retention
See "Mount Retention Period" on page 144.

Library
See "Library" on page 146.

Maximum Capacity

You can specify a maximum capacity value that restricts the size of volumes (that is, files) associated with a REMOVABLEFILE device class. Use the MAXCAPACITY parameter with the DEFINE DEVCLASS command.

Because the server opens only one file per physical removable medium, specify a value such that the one file makes full use of your media capacity. When the server detects that a volume has reached a size equal to the maximum capacity, it treats the volume as full and stores any new data on a different volume.

The default MAXCAPACITY value for a REMOVABLEFILE device class is the remaining space in the file system where the removable media volume is added to TSM.

You can update the device class information by issuing the UPDATE DEVCLASS command.

Defining and Updating FILE Device Classes

The FILE device type is used for storing data on disk in *simulated* storage volumes. The storage volumes are actually files. Data is written sequentially into standard files in the file system of the server machine. You can define this device class by issuing a DEFINE DEVCLASS command with the DEVTYPE=FILE parameter. Because each volume in a FILE device class is actually a file, a volume name must be a fully qualified file name.

Note: Do not use raw partitions with a device class type of FILE.

When you define or update the FILE device class, you can specify the parameters described in the following sections.

Mount Limit

The mount limit value for FILE device classes is used to restrict the number of mount points (volumes or files) that can be concurrently opened for access by server storage and retrieval operations. Any attempts to access more volumes than indicated by the mount limit causes the requester to wait. The default value is 1. The maximum value for this parameter is 256.

Note: The MOUNTLIMIT=DRIVES parameter is not valid for the FILE device class.

When selecting a mount limit for this device class, consider how many TSM processes you want to run at the same time.

TSM automatically cancels some processes to run other, higher priority processes. If the server is using all available mount points in a device class to complete higher priority processes, lower priority processes must wait until a mount point becomes available. For example, TSM cancels the process for a client backup if the mount point being used is needed for a server migration or reclamation process. TSM cancels a reclamation process if the mount point being used is needed for a client restore operation. For additional information, see “Preemption of Client or Server Operations” on page 361.

If processes are often cancelled by other processes, consider whether you can make more mount points available for TSM use. Otherwise, review your scheduling of operations to reduce the contention for resources.

Maximum Capacity Value

You can specify a maximum capacity value that restricts the size of volumes (that is, files) associated with a FILE device class. Use the MAXCAPACITY parameter of the DEFINE DEVCLASS command. When the server detects that a volume has reached a size equal to the maximum capacity, it treats the volume as full and stores any new data on a different volume.

The default MAXCAPACITY value for a FILE device class is 4MB.

Directory

You can specify the directory location of the files used in the FILE device class. The default is the current working directory of the server at the time the command is issued, unless the DSMSERV_DIR environment variable is set. For more information on setting the environment variable, refer to *Quick Start*.

The directory name identifies the location where the server places the files that represent storage volumes for this device class. While processing the command, the server expands the specified directory name into its fully qualified form, starting from the root directory.

Later, if the server needs to allocate a scratch volume, it creates a new file in this directory. The following lists the file name extension created by the server for scratch volumes depending on the type of data that is stored.

For scratch volumes used to store this data:	**The file extension is:**
Client data	.BFS
Export	.EXP
Database backup	.DBB
Database dump and unload	.DMP

Defining and Updating SERVER Device Classes

The SERVER device type is used for special device classes whose storage volumes are not directly attached to this server. A volume with device type SERVER consists of one or more files archived in the server storage of another server, called a target server. You can define this device class by issuing a DEFINE DEVCLASS command with the DEVTYPE=SERVER parameter. For information about how to use a SERVER device class, see “Using Virtual Volumes to Store Data on Another Server” on page 469.

The following parameters specify how to manage data storage operations for the new device class:

- SERVERNAME
- MOUNTLIMIT
- MAXCAPACITY
- MOUNTRETENTION
- PREFIX
- RETRYPERIOD
- RETRYINTERVAL

You can update the device class information by issuing the UPDATE DEVCLASS command.

Server Name

The TSM server on which you define a SERVER device class is called a source server. The source server uses the SERVER device class to store data on another TSM server, called a target server.

When defining a SERVER device class, specify the name of the target server. The target server must already be defined by using the DEFINE SERVER command. See “Using Virtual Volumes to Store Data on Another Server” on page 469 for more information.

Mount Limit

Use the mount limit value for SERVER device classes to restrict the number of simultaneous sessions between the source server and the target server. Any attempts to access more sessions than indicated by the mount limit causes the requester to wait. The default mount limit value is 1. The maximum value for this parameter is 256.

Note: The MOUNTLIMIT=DRIVES parameter is not valid for the SERVER device class.

When selecting a mount limit, consider your network load balancing and how many TSM processes you want to run at the same time.

TSM automatically cancels some processes to run other, higher priority processes. If the server is using all available sessions in a device class to complete higher priority processes, lower priority processes must wait until a session becomes available. For example, TSM cancels the process for a client backup if a session is needed for a server migration or reclamation process. TSM cancels a reclamation process if the session being used is needed for a client restore operation.

Also consider the resources available on the target server when setting mount limits. Do not set a high mount limit value if the target cannot move enough data or access enough data to satisfy all of the requests.

If processes are often cancelled by other processes, consider whether you can make more sessions available for TSM use. Otherwise, review your scheduling of operations to reduce the contention for network resources.

Maximum Capacity Value

You can specify a maximum capacity value that restricts the size of files that are created on the target server to store data for the source server. The default MAXCAPACITY value is 500MB. The storage pool volumes of this device type are explicitly set to full when the volume is closed and dismounted.

Mount Retention

You can specify the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. The default value is 60. The maximum value you can specify for this parameter is 9999. A value of 1 to 5 minutes is recommended. This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

Prefix

You can specify a prefix that the source server will use as the beginning portion of the high-level archive file name on the target server.

Retry Period

You can specify a retry period for communications with the target server. When there is a communications failure, this period determines the amount of time during which the source server continues to attempt to connect to the target server.

Retry Interval

You can specify how often the source server tries to connect to the target server when there is a communications failure. During the retry period, the source server tries to connect again as often as indicated by the retry interval.

Requesting Information about a Device Class

You can choose to view a standard or detailed report for a device class.

Task	Required Privilege Class
Request information about device classes	Any administrator

To display a standard report on device classes, enter:

```
query devclass
```

Figure 13 is an example of a standard report for device classes.

```
Device       Device        Storage    Device     Format     Est/Max    Mount
Class        Access           Pool    Type                 Capacity    Limit
Name         Strategy        Count                             (MB)
---------    ----------    -------    -------    ------    --------    -----
DISK         Random              9
TAPE8MM      Sequential          1    8MM        8200       2,472.0        2
```

Figure 13. Example of a Standard Device Class Report

To display a detailed report on the TAPE8MM device class, enter:

```
query devclass tape8mm format=detailed
```

Figure 14 shows an example of a detailed report for a device class.

```
             Device Class Name: TAPE8MM
        Device Access Strategy: Sequential
            Storage Pool Count: 1
                   Device Type: 8MM
                        Format: 8200
        Est/Max Capacity (MB): 2,472.0
                   Mount Limit: 2
              Mount Wait (min): 10
         Mount Retention (min): 30
                  Label Prefix: ADSM
                       Library: TAPELIB
                     Directory:
Last Update by (administrator): TSMADMIN
         Last Update Date/Time: 01/05/2001 16:02:13
```

Figure 14. Example of a Detailed Device Class Report

Deleting a Device Class

Task	Required Privilege Class
Delete a device classes	System or unrestricted storage

You can delete a device class with the DELETE DEVCLASS command when:

- No storage pools are assigned to the device class. For information on deleting storage pools, see “Deleting a Storage Pool” on page 221.
- The device class is not being used by an export or import process.

Note: You cannot delete the DISK device class from the server.

How TSM Fills Volumes

The DEFINE DEVCLASS command has an optional ESTCAPACITY parameter that indicates the estimated capacity for sequential volumes associated with the device class. If the ESTCAPACITY parameter is not specified, TSM uses a default value based on the recording format specified for the device class (FORMAT=).

If you specify an estimated capacity that exceeds the actual capacity of the volume in the device class, TSM updates the estimated capacity of the volume when the volume becomes full. When TSM reaches the end of the volume, it updates the capacity for the amount that is written to the volume.

You can either accept the default estimated capacity for a given device class, or explicitly specify an estimated capacity. An accurate estimated capacity value is not required, but is useful. TSM uses the estimated capacity of volumes to determine the estimated capacity of a storage pool, and the estimated percent utilized. You may want to change the estimated capacity if:

- The default estimated capacity is inaccurate because data compression is being performed by the drives
- You have volumes of nonstandard size

Using Data Compression

Client files can be compressed to decrease the amount of data sent over networks and the space occupied by the data in TSM storage. With TSM, files can be compressed by the TSM client before the data is sent to the TSM server, or by the device where the file is finally stored.

Use either client compression or device compression, but not both. The following table summarizes the advantages and disadvantages of each type of compression.

Type of Compression	Advantages	Disadvantages
TSM client compression	Reduced load on the network	Higher CPU usage by the client Longer elapsed time for client operations such as backup
Drive compression	Amount of compression can be better than TSM client compression on some drives	Files that have already been compressed by the TSM client can become larger

Either type of compression can affect tape drive performance, because compression affects data rate. When the rate of data going to a tape drive is slower than the drive can write, the drive starts and stops while data is written, meaning relatively poorer performance. When the rate of data is fast enough, the tape drive can reach streaming mode, meaning better performance. If tape drive performance is more important than the space savings that compression can mean, you may want to perform timed test backups using different approaches to determine what is best for your system.

Drive compression is specified with the FORMAT parameter for the drive's device class, and the hardware device must be able to support the compression format.

For information about how to set up compression on the client, see “Node Compression Considerations” on page 229 and “Registering Nodes with the Server” on page 228.

Tape Volume Capacity and Data Compression

How TSM views the capacity of the volume where the data is stored depends on whether files are compressed by the TSM client or by the storage device. It may wrongly appear that you are not getting the full use of the capacity of your tapes, for the following reasons:

- A tape device manufacturer often reports the capacity of a tape based on an assumption of compression by the device. If a client compresses a file before it is sent, the device may not be able to compress it any further before storing it.
- TSM records the size of a file as it goes to a storage pool. If the client compresses the file, TSM records this smaller size in the database. If the drive compresses the file, TSM is not aware of this compression.

Figure 15 on page 154 compares what TSM sees as the amount of data stored on tape when compression is done by the device and by the client. For this example, the tape has a physical capacity of 1.2 GB; however, the manufacturer reports the capacity of the tape as 2.4 GB by assuming the device compresses the data by a factor of two.

Suppose a client backs up a 2.4 GB file:

- When the client does *not* compress the file, the server records the file size as 2.4 GB, the file is compressed by the drive to 1.2 GB, and the file fills up one tape.
- When the client compresses the file, the server records the file size as 1.2 GB, the file cannot be compressed any further by the drive, and the file still fills one tape.

In both cases, TSM considers the volume to be full. However, TSM considers the capacity of the volume in the two cases to be different: 2.4 GB when the drive compresses the file, and 1.2 GB when the client compresses the file. Use the QUERY VOLUME command to see the capacity of volumes from TSM's viewpoint. See "Monitoring the Use of Storage Pool Volumes" on page 199.

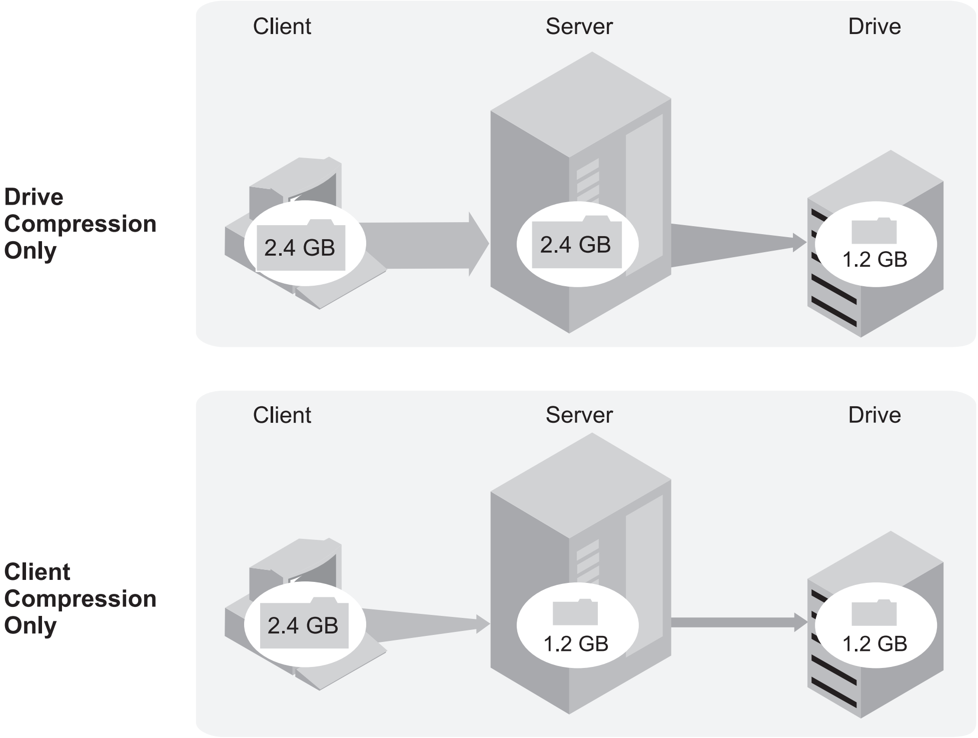


Figure 15. Comparing Compression at the Client and Compression at the Device

For how to set up compression on the client, see "Node Compression Considerations" on page 229 and "Registering Nodes with the Server" on page 228.

Chapter 9. Managing Storage Pools and Volumes

When you configure devices so that the Tivoli Storage Manager server can use them to store client data, you create storage pools and storage volumes. This section gives you overviews and details on storage pools and storage volumes.

The procedures in “Chapter 3. Using Magnetic Disk Devices” on page 51 and “Chapter 5. Configuring Storage Devices” on page 67 show you how to set up and use devices to provide TSM with server storage. The procedures use the set of defaults that TSM provides for storage pools and volumes. The defaults can work well, but you may have specific requirements not met by the defaults. Three common reasons to change the defaults are the following:

- Optimize and control storage device usage — Arrange the storage hierarchy and tune migration through the hierarchy (next storage pool, migration thresholds)
- Reuse tape volumes (reclamation) (Reuse is also related to policy and expiration.)
- Keep a client’s files on a minimum number of volumes (collocation)

You can also make other adjustments to tune the server for your systems. See the following sections to learn more. For some quick tips, see Table 13 on page 162.

Concepts:
“Overview: Storage Pools” on page 156
“Overview: Volumes in Storage Pools” on page 164
“Access Modes for Storage Pool Volumes” on page 168
“Overview: The Storage Pool Hierarchy” on page 170
“Migration of Files in a Storage Pool Hierarchy” on page 175
“Using Cache on Disk Storage Pools” on page 183
“Keeping a Client’s Files Together: Collocation” on page 184
“Reclaiming Space in Sequential Access Storage Pools” on page 189
“Estimating Space Needs for Storage Pools” on page 196

Tasks:
“Defining or Updating Primary Storage Pools” on page 158
“Preparing Volumes for Random Access Storage Pools” on page 166
“Preparing Volumes for Sequential Access Storage Pools” on page 166
“Defining Storage Pool Volumes” on page 167
“Updating Storage Pool Volumes” on page 167
“Setting Up a Storage Pool Hierarchy” on page 170
“Monitoring Storage Pools and Volumes” on page 198
“Monitoring the Use of Storage Pool Volumes” on page 199
“Moving Files from One Volume to Another Volume” on page 211
“Deleting a Storage Pool” on page 221
“Deleting Storage Pool Volumes” on page 222

© Copyright IBM Corp. 1993, 2002

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Overview: Storage Pools

A storage volume is the basic unit of storage, such as allocated space on a disk or a single tape cartridge. A storage pool is a collection of storage volumes. The server uses the storage volumes to store backed-up, archived, or space-managed files. The group of storage pools that you set up for the TSM server to use is called *server storage*. Storage pools can be arranged in a storage hierarchy.

The server has two types of storage pools that serve different purposes: primary storage pools and copy storage pools.

Primary Storage Pool

When a user tries to restore, retrieve, recall, or export file data, the requested file is obtained from a primary storage pool if possible. Primary storage pool volumes are always located onsite.

A primary storage pool can use random access storage (DISK device class) or sequential access storage (for example, tape or FILE device classes).

The server has three default, random access, primary storage pools:

ARCHIVEPOOL
In default STANDARD policy, the destination for files that are archived from client nodes

BACKUPPOOL
In default STANDARD policy, the destination for files that are backed up from client nodes

SPACEMGPOOL
For space-managed files that are migrated from Tivoli Space Manager client nodes (HSM clients)

The server does not require separate storage pools for archived, backed-up, or space-managed files. However, you may want to have a separate storage pool for space-managed files. Clients are likely to require fast access to their space-managed files. Therefore, you may want to have those files stored in a separate storage pool that uses your fastest disk storage.

Copy Storage Pool

When an administrator backs up a primary storage pool, the data is stored in a copy storage pool. See "Backing Up Storage Pools" on page 507 for details.

A copy storage pool can use only sequential access storage (for example, a tape device class or FILE device class).

The copy storage pool provides a means of recovering from disasters or media failures. For example, when a client attempts to retrieve a file and the server detects an error in the file copy in the primary storage pool, the server marks the file as damaged. At the next attempt to access the file, the server obtains the file from a copy storage pool. For details, see "Restoring Storage Pools" on page 525, "Using Copy Storage Pools to Improve Data Availability" on page 510, "Recovering a Lost or Damaged Storage Pool Volume" on page 543, and "Maintaining the Integrity of Files" on page 537.

You can move copy storage pool volumes offsite and still have the server track the volumes. Moving copy storage pool volumes offsite provides a means of recovering from an onsite disaster.

Note: To back up a primary storage pool the DATAFORMAT must be NATIVE or NONBLOCK.

An Example of Server Storage

Figure 16 shows one way to set up server storage. In this example, the storage defined for the server includes:

- Three disk storage pools, which are primary storage pools: ARCHIVE, BACKUP, and HSM
- One primary storage pool that consists of tape cartridges
- One copy storage pool that consists of tape cartridges

Policies defined in management classes direct the server to store files from clients in the ARCHIVE, BACKUP, or HSM disk storage pools. For each of the three disk storage pools, the tape primary storage pool is next in the hierarchy. As the disk storage pools fill, the server migrates files to tape to make room for new files. Large files may go directly to tape. For more information about setting up a storage hierarchy, see "Overview: The Storage Pool Hierarchy" on page 170.

You can back up all four of the primary storage pools to the one copy storage pool. For more information on backing up primary storage pools, see "Backing Up Storage Pools" on page 507.

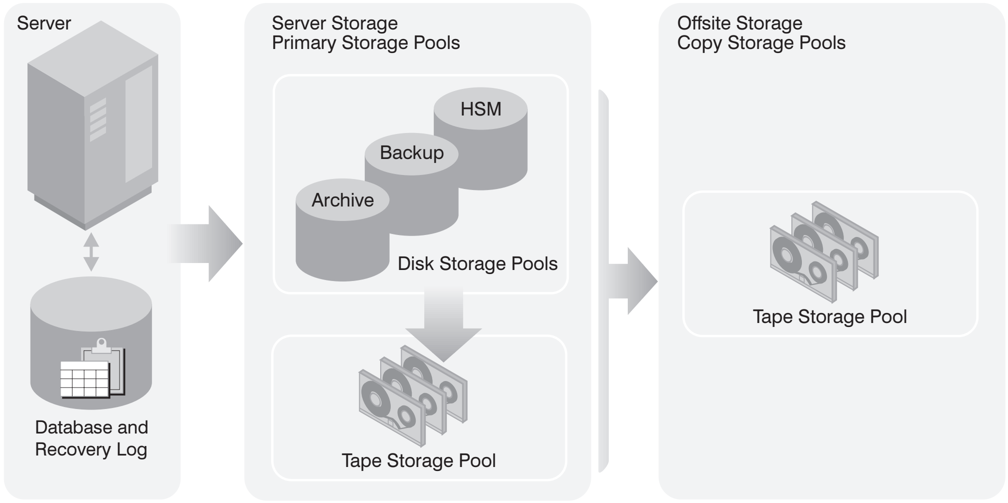


Figure 16. Example of Server Storage

To set up this server storage hierarchy, do the following:

1. Define the three disk storage pools, or use the three default storage pools that are defined when you install the server. Add volumes to the disk storage pools if you have not already done so.

   See “Configuring Random Access Volumes on Disk Devices” on page 52.
2. Define policies that direct the server to initially store files from clients in the disk storage pools. To do this, you define or change management classes and copy groups so that they point to the storage pools as destinations. Then activate the changed policy. See “Changing Policy” on page 272 for details.
3. Attach one or more tape devices, or a tape library, to your server system.

   To enable the server to use the device, you must enter a series of the following commands:

   ```
   DEFINE LIBRARY
   DEFINE DRIVE
   DEFINE DEVCLASS
   DEFINE STGPOOL
   ```

   See “Chapter 5. Configuring Storage Devices” on page 67 for more information. For detailed information on defining a storage pool, see “Defining or Updating Primary Storage Pools”.
4. Update the disk storage pools so that they point to the tape storage pool as the next storage pool in the hierarchy. See “Example: Updating Storage Pools” on page 162.
5. Define a copy storage pool. This storage pool can use the same tape device or a different tape device as the primary tape storage pool. See “Defining a Copy Storage Pool” on page 218
6. Set up administrative schedules or a script to back up the disk storage pools and the tape storage pool to the copy storage pool. Send the volumes offsite for safekeeping. See “Backing Up Storage Pools” on page 507.

Defining or Updating Primary Storage Pools

This section provides a summary of parameters you can set and change for storage pools using the administrative command-line or the administrative Web interface. The section also provides examples of defining and updating storage pools.

Task	Required Privilege Class
Define storage pools	System
Update storage pools	System or unrestricted storage

When you define a primary storage pool, be prepared to provide some or all of the information that is shown in Table 12. Most of the information is optional. Some information applies only to random access storage pools or only to sequential access storage pools. Required parameters are marked.

Table 12. Information for Defining a Storage Pool

Information	Explanation	Type of Storage Pool
Storage pool name *(Required)*	The name of the storage pool.	random, sequential
Device class *(Required)*	The name of the device class assigned for the storage pool.	random, sequential

Table 12. Information for Defining a Storage Pool (continued)

Information	Explanation	Type of Storage Pool
Pool type	The type of storage pool (primary or copy). The default is to define a primary storage pool. Once you define a storage pool, you cannot change whether it is a primary or a copy storage pool.	random, sequential
Maximum number of scratch volumes *(Required for sequential access)*	When you specify a value greater than zero, the server dynamically acquires scratch volumes when needed, up to this maximum number. For automated libraries, set this value equal to the physical capacity of the library. See "Maintaining a Supply of Scratch Volumes in an Automated Library" on page 127.	sequential
Access mode	Defines access to volumes in the storage pool for user operations (such as backup and restore) and system operations (such as reclamation and server migration). Possible values are: **Read/Write** User and system operations can read from or write to the volumes. **Read-Only** User operations can read from the volumes, but not write. Server processes can move files within the volumes in the storage pool. However, no new writes are permitted to volumes in the storage pool from volumes outside the storage pool. **Unavailable** User operations cannot get access to volumes in the storage pool. No new writes are permitted to volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool.	random, sequential
Maximum file size	To exclude large files from a storage pool, set a maximum file size. The maximum file size applies to the size of a physical file (a single client file or an aggregate of client files). Do not set a maximum file size for the last storage pool in the hierarchy unless you want to exclude very large files from being stored in server storage.	random, sequential
Cyclic Redundancy Check (CRC)	Specifies whether the server uses CRC to validate storage pool data during audit volume processing. For additional information see "Data Validation During Audit Volume Processing" on page 531. Using the CRC option in conjunction with scheduling audit volume processing continually ensures the integrity of data stored in your storage hierarchy. If you always want your storage pool data validated, set you primary storage pool crcdata definition to YES.	random, sequential
Name of the next storage pool	Specifies the name of the next storage pool in the storage pool hierarchy, where files can be migrated or stored. See "Overview: The Storage Pool Hierarchy" on page 170. When defining copy storage pools to primary pools that have defined next pools, the copy pool list for each primary pool should be the same. Defining different copy pool lists can cause resources to be freed when failing over to the next pool. If the resources are freed, it can delay the completion of client operations.	random, sequential
Migration thresholds	Specifies a percentage of storage pool occupancy at which the server begins migrating files to the next storage pool (high threshold) and the percentage when migration stops (low threshold). See "Migration of Files in a Storage Pool Hierarchy" on page 175.	random, sequential

Table 12. Information for Defining a Storage Pool (continued)

Information	Explanation	Type of Storage Pool
Migration processes	Specifies the number of processes that are used for migrating files from this storage pool. See "Migration for Disk Storage Pools" on page 176.	random
Migration delay	Specifies whether migration of files should be delayed for a minimum number of days. See "Keeping Files in a Storage Pool" on page 179 and "How Tivoli Storage Manager Migrates Data from Sequential Access Storage Pools" on page 181.	random, sequential
Continue migration process	Specifies whether migration of files should continue even if files do not meet the requirement for migration delay. This setting is used only when the storage pool cannot go below the low migration threshold without moving additional files. See "Keeping Files in a Storage Pool" on page 179 and "How Tivoli Storage Manager Migrates Data from Sequential Access Storage Pools" on page 181.	random, sequential
Cache	Enables or disables cache. When cache is enabled, copies of files migrated by the server to the next storage pool are left on disk after the migration. In this way, a retrieval request can be satisfied quickly. See "Using Cache on Disk Storage Pools" on page 183.	random
Collocation	With collocation enabled, the server attempts to keep all files belonging to a client node or a client file space on a minimal number of sequential access storage volumes. See "Keeping a Client's Files Together: Collocation" on page 184.	sequential
Reclamation threshold	Specifies what percentage of reclaimable space can accumulate on a volume before the server initiates a space reclamation process for the volume. See "Choosing a Reclamation Threshold" on page 191.	sequential
Reclamation storage pool	Specifies the name of the storage pool to be used for storing data from volumes being reclaimed in this storage pool. Use for storage pools whose device class only has one drive or mount point. See "Reclaiming Volumes in a Storage Pool with One Drive" on page 192.	sequential
Reuse delay period	Specifies the number of days that must elapse after all of the files have been deleted from a volume, before the volume can be rewritten or returned to the scratch pool. See "Delaying Reuse of Sequential Access Volumes" on page 509.	sequential
Overflow location	Specifies the name of a location where volumes are stored when they are ejected from an automated library by the MOVE MEDIA command. Use for a storage pool that is associated with an automated library or an external library. See "Managing a Full Library" on page 125.	sequential
Data Format	The format in which data will be stored. NATIVE is the default data format. NETAPPDUMP and NONBLOCK are examples of other data formats. NETAPPDUMP is a format used when you use Tivoli Data Protection for NDMP. For more information about NDMP, see "Chapter 6. Setting Up Tivoli Data Protection for NDMP" on page 97.	sequential

Table 12. Information for Defining a Storage Pool (continued)

Information	Explanation	Type of Storage Pool
Copy Storage Pools	Specifies the names of copy storage pools where copies of files being stored in the primary storage pool, during client backup, archive or HSM stores, are also simultaneously written to all listed copy storage pools. This option is restricted to only primary storage pools using NATIVE or NONBLOCK data format. See the Copy Continue entry and "Simultaneous Write to a Primary Storage Pool and Copy Storage Pools" on page 163 for related information. **Notes:** 1. The COPYSTGPOOLS parameter is not intended to replace the BACKUP STGPOOL command. If you use the copy storage pools function, ensure that the copy of the storage pool is complete by using the BACKUP STGPOOL command. 2. When defining copy storage pools to primary pools that have defined next pools, the copy pool list for each primary pool should be the same. Defining different copy pool lists can cause resources to be freed when failing over to the next pool. If the resources are freed, it can delay the completion of client operations.	sequential
Copy Continue	Specifies how the server should react to a copy storage pool write failure for any of the copy storage pools listed in the COPYSTGPOOLS parameter. With a value of YES, during a write failure, the server will exclude the failing copy storage pool from any further writes while that specific client session is active. With a value of NO, during a write failure, the server will fail the entire transaction including the write to the primary storage pool.	sequential

Example: Defining Storage Pools

For this example, suppose you have determined that an engineering department requires a separate storage hierarchy. You want the department's backed-up files to go to a disk storage pool. When that pool fills, you want the files to migrate to a tape storage pool. You want the pools to have the following characteristics:

- Disk primary storage pool
  - The pool named ENGBACK1 is the storage pool for the engineering department.
  - The size of the largest file that can be stored is 5MB. Files larger than 5MB are stored in the tape storage pool.
  - Files migrate from the disk storage pool to the tape storage pool when the disk storage pool is 85% full. File migration to the tape storage pool stops when the disk storage pool is down to 40% full.
  - The access mode is the default, read/write.
  - Cache is used.
- Tape primary storage pool
  - The name of the pool is BACKTAPE.
  - The pool uses the device class TAPE, which has already been defined.
  - No limit is set for the maximum file size, because this is the last storage pool in the hierarchy.
  - To group files from the same client on a small number of volumes, use collocation at the client node level.
  - Use scratch volumes for this pool, with a maximum number of 100 volumes.
  - The access mode is the default, read/write.

- Use the default for reclamation: Reclaim a partially full volume (to allow tape reuse) when 60% of the volume's space can be reclaimed.

You can define the storage pools in a storage pool hierarchy from the top down or from the bottom up. Defining the hierarchy from the bottom up requires fewer steps. To define the hierarchy from the bottom up, perform the following steps:

1. Define the storage pool named BACKTAPE with the following command:

   ```
   define stgpool backtape tape
   description='tape storage pool for engineering backups'
   maxsize=nolimit collocate=yes maxscratch=100
   ```

2. Define the storage pool named ENGBACK1 with the following command:

   ```
   define stgpool engback1 disk
   description='disk storage pool for engineering backups'
   maxsize=5m nextstgpool=backtape highmig=85 lowmig=40
   ```

Restrictions:

1. You cannot establish a chain of storage pools that lead to an endless loop. For example, you cannot define StorageB as the *next* storage pool for StorageA, and then define StorageA as the *next* storage pool for StorageB.
2. The storage pool hierarchy includes only primary storage pools, not copy storage pools.
3. If a storage pool uses the data format NETAPPDUMP, the server will not perform storage pool backup, migration, reclamation, MOVE DATA, and AUDIT VOLUME on that storage pool. For more information on the NETAPPDUMP data format, see "Chapter 6. Setting Up Tivoli Data Protection for NDMP" on page 97.

Example: Updating Storage Pools

You can update storage pools to change the storage hierarchy and other characteristics.

For example, suppose you had already defined the ENGBACK1 disk storage pool according to the previous example. Now you have decided to increase the maximum size of a physical file that may be stored in the storage pool. Use the following command:

```
update stgpool engback1 maxsize=100m
```

Note:

- You cannot use this command to change the data format for a storage pool.
- For storage pools that have the NETAPPDUMP data format, you can modify only the following parameters: DESCRIPTION, ACCESS, COLLOCATE, MAXSCRATCH, REUSEDELAY.

Task Tips for Storage Pools

Table 13 gives tips on how to accomplish some tasks that are related to storage pools.

Table 13. Task Tips for Storage Pools

For this Goal	Do This	For More Information
Keep the data for a client on as few volumes as possible	Enable collocation for the storage pool	"Keeping a Client's Files Together: Collocation" on page 184

Table 13. Task Tips for Storage Pools (continued)

For this Goal	Do This	For More Information
Reduce the number of volume mounts needed to back up multiple clients	Disable collocation for the storage pool	"Keeping a Client's Files Together: Collocation" on page 184
Perform a simultaneous write to a primary storage pool and copy storage pools	Provide a list of copy storage pools when defining the primary storage pool.	"Simultaneous Write to a Primary Storage Pool and Copy Storage Pools"
Specify how the server reuses tapes	Set a reclamation threshold for the storage pool Optional: Identify a reclamation storage pool	"Reclaiming Space in Sequential Access Storage Pools" on page 189
Move data from disk to tape automatically when needed	Set a migration threshold for the storage pool Identify the next storage pool	"Migration for Disk Storage Pools" on page 176
Move data from disk to tape automatically based on how frequently users access the data or how long the data has been in the storage pool	Set a migration threshold for the storage pool Identify the next storage pool Set the migration delay period	"Migration for Disk Storage Pools" on page 176
Back up your storage pools	Define a copy storage pool Set up a backup schedule	"Defining a Copy Storage Pool" on page 218 "Automating a Basic Administrative Command Schedule" on page 367
Have clients back up directly to a tape storage pool	Define a sequential access storage pool that uses a tape device class Change the policy that the clients use, so that the backup copy group points to the tape storage pool as the destination.	"Defining or Updating Primary Storage Pools" on page 158 "Changing Policy" on page 272

Simultaneous Write to a Primary Storage Pool and Copy Storage Pools

To simultaneously write data to a primary storage pool and one or more copy storage pools, you can specify a list of copy storage pools in a primary storage pool definition using the COPYSTGPOOLS parameter. When a client backs up, archives, or migrates a file, the file is written to the primary storage pool and is simultaneously stored into each copy storage pool. If a write failure occurs for any of the copy storage pools, the COPYCONTINUE parameter setting determines how the server should react. See Table 12 on page 158 for additional information about storage pool definitions.

When using copy storage pools for the simultaneous write of data to a primary storage pool and associated copy storage pools, be sure that all resources, such as enough drives, are available for the write operation. For example, each simultaneous write operation will need all of the volumes to be simultaneously mounted. This requires that the number of available drives must be equal to or greater than the number of volumes to be simultaneously mounted. For additional information, see "Mount Limit" on page 143 and the REGISTER NODE command in the *Administrator's Reference* for information about the MAXNUMMP parameter.

Note: When defining copy storage pools to primary pools that have defined next pools, the copy pool list for each primary pool should be the same. Defining different copy pool lists can cause resources to be freed when using the next pool. If the resources are freed, it can delay the completion of client operations.

Overview: Volumes in Storage Pools

Storage pool volumes are the physical media that are assigned to a storage pool. Some examples of volumes are:

- Space allocated on a disk drive
- A tape cartridge
- An optical disk

Storage pools and their volumes are either random access or sequential access, depending on the device type of the device class to which the pool is assigned.

Random Access Storage Pool Volumes

Random access storage pools consist of volumes on disk. Random access storage pools are always associated with the DISK device class, and all volumes are one of the following:

- Fixed-size files on a disk. The files are created when you define volumes.
- Raw logical volumes that must be defined, typically by using SMIT, before the server can access them.

  Attention: It is recommended that you use journal file system (JFS) files rather than raw logical volumes for storage pool volumes. See "The Advantages of Using Journal File System Files" on page 385 for details.

See "Preparing Volumes for Random Access Storage Pools" on page 166 for details.

Sequential Access Storage Pool Volumes

Volumes in sequential access storage pools include any supported device type to which the server writes data sequentially. Some examples of sequential access volumes are:

- Tape cartridge
- Optical disk
- File

Each volume defined in a sequential access storage pool must be of the same type as the device type of the associated device class. See Table 14 for the type of volumes associated with each device type.

For preparing sequential access volumes, see "Preparing Volumes for Sequential Access Storage Pools" on page 166.

Table 14. Volume Types

Device Type	Volume Description	Label Required
3570	IBM 3570 tape cartridge	Yes
3590	IBM 3590 tape cartridge	Yes
4MM	4mm tape cartridge	Yes
8MM	8mm tape cartridge	Yes

Table 14. Volume Types (continued)

Device Type	Volume Description	Label Required
CARTRIDGE	IBM 3480 or 3490 cartridge system tape	Yes
DLT	A digital linear tape	Yes
DTF	A digital tape format (DTF) tape	Yes
ECARTRIDGE	A cartridge tape that is used by a tape drive such as the StorageTek SD-3 or 9490 tape drive	Yes
FILE	A file in the file system of the server machine	No
GENERICTAPE	A tape that is compatible with the drives that are defined to the device class	Yes
NAS	A network-attached storage (NAS) device is defined to the device class	Yes
OPTICAL	A two-sided 5.25-inch rewritable optical cartridge	Yes
QIC	A 1/4-inch tape cartridge	Yes
REMOVABLEFILE	A file on a removable medium. If the medium has two sides, each side is a separate volume.	Yes
SERVER	One or more objects that are archived in the server storage of another server	No
WORM	A two-sided 5.25-inch write-once optical cartridge	Yes
WORM12	A two-sided 12-inch write-once optical cartridge	Yes
WORM14	A two-sided 14-inch write-once optical cartridge	Yes

Scratch Volumes Versus Defined Volumes

You can define volumes in a sequential access storage pool or you can specify that the server dynamically acquire scratch volumes. You can also use a combination of defined and scratch volumes. What you choose depends on the amount of control you need over individual volumes.

Use defined volumes when you want to control precisely which volumes are used in the storage pool. Using defined volumes may be useful when you want to establish a naming scheme for volumes.

Use scratch volumes to enable the server to define a volume when needed and delete the volume when it becomes empty. Using scratch volumes frees you from the burden of explicitly defining all of the volumes in a storage pool.

The server tracks whether a volume being used was originally a scratch volume. Scratch volumes that the server acquired for a primary storage pool are deleted from the server database when they become empty. The volumes are then available for reuse by the server or other applications. For scratch volumes that were acquired in a FILE device class, the space that the volumes occupied is freed by the server and returned to the file system.

Scratch volumes in a copy storage pool are handled in the same way as scratch volumes in a primary storage pool, except for volumes with the access value of offsite. If an offsite volume becomes empty, the server does not immediately return the volume to the scratch pool. The delay prevents the empty volumes from being deleted from the database, making it easier to determine which volumes should be returned to the onsite location. The administrator can query the server for empty offsite copy storage pool volumes and return them to the onsite location. The volume is returned to the scratch pool only when the access value is changed to READWRITE, READONLY, or UNAVAILABLE.

Preparing Volumes for Random Access Storage Pools

For a random access storage pool, you must define volumes.

Task	Required Privilege Class
Define volumes in any storage pool	System or unrestricted storage
Define volumes in specific storage pools	System, unrestricted storage, or restricted storage for those pools

Prepare a volume for use in a random access storage pool by defining the volume. For example, suppose you want to define a 21MB volume for the BACKUPPOOL storage pool. You want the volume to be located in the path */usr/lpp/adsmserv/bin* and named stgvol.001. Enter the following command:

```
define volume backuppool /usr/lpp/adsmserv/bin/stgvol.001 formatsize=21
```

If you do not specify a full path name for the volume name, the command uses the current path.

Tip: Define storage pool volumes on disk drives that reside on the TSM server machine, not on remotely mounted file systems.

Note: This one-step process replaces the former two-step process of first formatting a volume (using DSMFMT) and then defining the volume. If you choose to use the two-step process, the DSMFMT utility is available from the operating system command line. See *Administrator's Reference* for details.

Another option for preparing a volume is to create a raw logical volume by using SMIT.

Preparing Volumes for Sequential Access Storage Pools

For sequential access storage pools with a FILE or SERVER device type, no labeling or other preparation of volumes is necessary.

For sequential access storage pools with other than a FILE or SERVER device type, you must prepare volumes for use. When the server accesses a sequential access volume, it checks the volume name in the header to ensure that the correct volume is being accessed. To prepare a volume:

1. Label the volume. Table 14 on page 164 shows the types of volumes that require labels. You must label those types of volumes before the server can use them.

   See "Labeling Removable Media Volumes" on page 114.

   Tip: When you use the LABEL LIBVOLUME command with drives in an automated library, you can label and check in the volumes with one command.
2. For storage pools in automated libraries, use the CHECKIN LIBVOLUME command to check the volume into the library. See "Checking New Volumes into a Library" on page 117.
3. If you have not allowed scratch volumes in the storage pool, you must identify the volume, by name, to the server. For details, see "Defining Storage Pool Volumes" on page 167.

   If you allowed scratch volumes in the storage pool by specifying a value greater than zero for the MAXSCRATCH parameter, you can let the server use scratch volumes, identify volumes by name, or do both. See “Using Scratch Volumes” for information about scratch volumes.

Defining Storage Pool Volumes

Task	Required Privilege Class
Define volumes in any storage pool	System or unrestricted storage
Define volumes in specific storage pools	System, unrestricted storage, or restricted storage for those pools

When you define a storage pool volume, you inform the server that the volume is available for storing backup, archive, or space-managed data.

For a sequential access storage pool, the server can use dynamically acquired scratch volumes, volumes that you define, or a combination.

To define a volume named VOL1 in the ENGBACK3 tape storage pool, enter:

```
define volume engback3 vol1
```

Using Scratch Volumes

You do not have to define volumes in sequential access storage pools if you allow storage pools to use scratch volumes. Use the MAXSCRATCH parameter when you define or update the storage pool. Setting the MAXSCRATCH parameter to a value greater than zero lets the storage pool dynamically acquire volumes as needed. The server automatically defines the volumes as they are acquired. The server also automatically deletes scratch volumes from the storage pool when the server no longer needs them.

Before the server can use a scratch volume with a device type other than FILE or SERVER, the volume must have a standard label. See “Preparing Volumes for Sequential Access Storage Pools” on page 166.

Updating Storage Pool Volumes

Task	Required Privilege Class
Update volumes	System or operator

You can update the attributes of a storage pool volume assigned to a primary or copy storage pool. Update a volume to:

- Reset any error state for a volume, by updating the volume to an access mode of read/write.
- Change the access mode of a volume, for example if a tape cartridge is moved offsite (offsite access mode) or damaged (destroyed access mode). See “Access Modes for Storage Pool Volumes” on page 168.
- Change the location for a volume in a sequential access storage pool.

An example of when to use the UPDATE VOLUME command is if you accidentally damage a volume. You can change the access mode to unavailable so that the server does not try to write or read data from the volume. For example, if the volume name is VOL1, enter the following command:

```
update volume vol1 access=unavailable
```

When using the UPDATE VOLUME command, be prepared to supply some or all of the information shown in Table 15.

Table 15. Information for Updating a Storage Pool Volume

Information	Explanation
Volume name *(Required)*	Specifies the name of the storage pool volume to be updated. You can specify a group of volumes to update by using wildcard characters in the volume name. You can also specify a group of volumes by specifying the storage pool, device class, current access mode, or status of the volumes you want to update. See the parameters that follow.
New access mode	Specifies the new access mode for the volume (how users and server processes such as migration can access files in the storage pool volume). See "Access Modes for Storage Pool Volumes" for descriptions of access modes. A random access volume must be varied offline before you can change its access mode to *unavailable* or *destroyed*. To vary a volume offline, use the VARY command. See "Varying Disk Volumes Online or Offline" on page 53. If a scratch volume that is empty and has an access mode of offsite is updated so that the access mode is read/write, read-only, or unavailable, the volume is deleted from the database.
Location	Specifies the location of the volume. This parameter can be specified only for volumes in sequential access storage pools.
Storage pool	Restricts the update to volumes in the specified storage pool.
Device class	Restricts the update to volumes in the specified device class.
Current access mode	Restricts the update to volumes that currently have the specified access mode.
Status	Restricts the update to volumes with the specified status (online, offline, empty, pending, filling, or full).
Preview	Specifies whether you want to preview the update operation without actually performing the update.

Access Modes for Storage Pool Volumes

Access to any volume in a storage pool is determined by the access mode assigned to that volume. You can change the access mode of a volume. The server can also change the access mode based on what happens when it tries to access a volume. For example, if the server cannot write to a volume having read/write access mode, the server automatically changes the access mode to read-only.

The access modes are:

Read/write

Allows files to be read from or written to a volume in the storage pool.

If the server cannot write to a read/write access volume, the server automatically changes the access mode to read-only.

If a scratch volume that is empty and has an access mode of offsite is updated so that the access mode is read/write, the volume is deleted from the database.

Read-only

Allows files to be read from but not written to a disk or tape volume.

If a scratch volume that is empty and has an access mode of offsite is updated so that the access mode is read-only, the volume is deleted from the database.

Unavailable

Specifies that the volume is not available for any type of access by the server.

You must vary offline a random access volume before you can change its access mode to *unavailable*. To vary a volume offline, use the VARY command. See "Varying Disk Volumes Online or Offline" on page 53.

If a scratch volume that is empty and has an access mode of offsite is updated so that the access mode is unavailable, the volume is deleted from the database.

Destroyed

Specifies that a primary storage pool volume has been permanently damaged. Neither users nor system processes (like migration) can access files stored on the volume.

This access mode is used to indicate an entire volume that should be restored using the RESTORE STGPOOL or RESTORE VOLUME command. After all files on a destroyed volume are restored to other volumes, the destroyed volume is automatically deleted from the database. See "How Restore Processing Works" on page 500 for more information.

Only volumes in primary storage pools can be updated to an access mode of destroyed.

You must vary offline a random access volume before you can change its access mode to *destroyed*. To vary a volume offline, use the VARY command. See "Varying Disk Volumes Online or Offline" on page 53. Once you update a random access storage pool volume to destroyed, you cannot vary the volume online without first changing the access mode.

If you update a sequential access storage pool volume to destroyed, the server does not attempt to mount the volume.

If a volume contains no files and the UPDATE VOLUME command is used to change the access mode to destroyed, the volume is deleted from the database.

Offsite

Specifies that a copy storage pool volume is at an offsite location and therefore cannot be mounted. Use this mode to help you track volumes that are offsite. The server treats offsite volumes differently, as follows:

- Mount requests are not generated for offsite volumes
- Data can be reclaimed or moved from offsite volumes by retrieving files from other storage pools
- Empty, offsite scratch volumes are not deleted from the copy storage pool

You can only update volumes in a copy storage pool to offsite access mode. Volumes that have the device type of SERVER (volumes that are actually archived objects stored on another TSM server) cannot have an access mode of offsite.

Overview: The Storage Pool Hierarchy

You can set up your devices so that the server automatically moves data from one device to another, or one media type to another. The selection can be based on characteristics such as file size or storage capacity. To do this, you set up different primary storage pools to form a storage pool hierarchy. A typical implementation may have a disk storage pool with a subordinate tape storage pool. When a client backs up a file, the server may initially store the file on disk according to the policy for that file. Later, the server may move the file to tape when the disk becomes full. This action by the server is called migration. You can also place a size limit on files that are stored on disk, so that large files are stored initially on tape instead of on disk.

For example, your fastest devices are disks, but you do not have enough space on these devices to store all data that needs to be backed up over the long term. You have tape drives, which are slower to access, but have much greater capacity. You can define a hierarchy so that files are initially stored on the fast disk volumes in one storage pool. This provides clients with quick response to backup requests and some recall requests. As the disk storage pool becomes full, the server migrates, or moves, data to volumes in the tape storage pool.

Migration of files from disk to sequential storage pool volumes is particularly useful because the server migrates all the files for a single node together. This gives you partial collocation for clients. Migration of files is especially helpful if you decide not to enable collocation for sequential storage pools. See “Keeping a Client’s Files Together: Collocation” on page 184 for details.

Setting Up a Storage Pool Hierarchy

You can set up a storage pool hierarchy when you first define storage pools. You can also change the storage pool hierarchy later.

You establish a hierarchy by identifying the *next* storage pool, sometimes called the subordinate storage pool. The server migrates data to the next storage pool if the original storage pool is full or unavailable. See “Migration of Files in a Storage Pool Hierarchy” on page 175 for detailed information on how migration between storage pools works.

Restrictions:

1. You cannot establish a chain of storage pools that leads to an endless loop. For example, you cannot define StorageB as the *next* storage pool for StorageA, and then define StorageA as the *next* storage pool for StorageB.
2. The storage pool hierarchy includes only primary storage pools, not copy storage pools. See “Using Copy Storage Pools to Back Up a Storage Hierarchy” on page 174.
3. A storage pool must use the NATIVE or NONBLOCK data formats to be part of a storage pool hierarchy. For example, a storage pool using the NETAPPDUMP data format cannot be part of a storage pool hierarchy.

Example: Defining a Storage Pool Hierarchy

For this example, suppose that you have determined that an engineering department requires a separate storage hierarchy. You set up policy so that the server initially stores backed up files for this department to a disk storage pool. When that pool fills, you want the server to migrate files to a tape storage pool. You want the pools to have the following characteristics:

- Primary storage pool on disk

- Name the storage pool ENGBACK1.
- Limit the size of the largest file that can be stored to 5MB. The server stores files that are larger than 5MB in the tape storage pool.
- Files migrate from the disk storage pool to the tape storage pool when the disk storage pool is 85% full. File migration to the tape storage pool stops when the disk storage pool is down to 40% full.
- Use caching, so that migrated files stay on disk until the space is needed for other files.

- Primary storage pool on tape
  - Name the storage pool BACKTAPE.
  - Use the device class TAPE, which has already been defined, for this storage pool.
  - Do not set a limit for the maximum file size, because this is the last storage pool in the hierarchy.
  - Use scratch volumes for this pool, with a maximum number of 100 volumes.

You can define the storage pools in a storage pool hierarchy from the top down or from the bottom up. Defining the hierarchy from the bottom up requires fewer steps. To define the hierarchy from the bottom up, perform the following steps:

1. Define the storage pool named BACKTAPE with the following command:
   ```
   define stgpool backtape tape
   description='tape storage pool for engineering backups'
   maxsize=nolimit collocate=yes maxscratch=100
   ```
2. Define the storage pool named ENGBACK1 with the following command:
   ```
   define stgpool engback1 disk
   description='disk storage pool for engineering backups'
   maxsize=5M nextstgpool=backtape highmig=85 lowmig=40
   ```

Example: Updating a Storage Pool Hierarchy

If you have already defined the storage pool at the top of the hierarchy, you can update the storage hierarchy to include a new storage pool.

For example, suppose that you had already defined the ENGBACK1 disk storage pool. Now you have decided to set up a tape storage pool to which files from ENGBACK1 can migrate. Perform the following steps to define the new tape storage pool and update the hierarchy:

1. Define the storage pool named BACKTAPE with the following command:
   ```
   define stgpool backtape tape
   description='tape storage pool for engineering backups'
   maxsize=nolimit collocate=yes maxscratch=100
   ```
2. Specify that BACKTAPE is the next storage pool defined in the storage hierarchy for ENGBACK1. To update ENGBACK1, enter:
   ```
   update stgpool engback1 nextstgpool=backtape
   ```

How the Server Groups Files before Storing

When a user backs up or archives files from a client node, the server may group multiple client files into an *aggregate* (a single physical file). The size of the aggregate depends on the sizes of the client files being stored, and the number of bytes and files allowed for a single transaction. Two options, one in the server options file and one in the client options file, affect the number of bytes and files allowed for a single transaction:

- The TXNGROUPMAX option in the server options file indicates the maximum number of logical files (client files) that a client may send to the server in a single transaction. The server can tune this option automatically if you set the SELFTUNETXNSIZE option to YES. The server then uses the value that is specified for TXNGROUPMAX in the server options file and adjusts it for each node until it obtains the best performance. It is recommended that the client specify the maximum transaction byte limit in the client options and rely on the automatic tuning.

Note: Although the values of TXNGROUPMAX, MOVEBATCHSIZE, and MOVESIZETHRESH may be changed, the settings in the server options file are not changed. Issuing a QUERY OPTION command displays only what is set in the server options file.

- The TXNBYTELIMIT option in the client options file indicates the total number of bytes that the client can send to the server in a single transaction.

  This option sets a target size for the aggregate file. An aggregate file will usually be smaller than the value specified by the TXNBYTELIMIT option. A logical file (a single user's file) that is larger than the value specified by TXNBYTELIMIT option will not become part of an aggregate, but will be stored as a single physical file.

  The recommended value is 25600.

Together these options allow you to control the size of aggregate files stored by the server.

When a Tivoli Space Manager client (HSM client) migrates files to the server, the files are not grouped into an aggregate.

Where the Files Are Stored

When a user backs up, archives, or migrates a file from a client node, the server looks at the management class that is bound to the file. The management class specifies the destination, the storage pool in which to store the file. The server then checks that storage pool to determine the following:

- If it is possible to write file data to the storage pool (access mode).
- If the size of the physical file exceeds the maximum file size allowed in the storage pool. For backup and archive operations, the physical file may be an aggregate file or a single client file.
- Whether sufficient space is available on the available volumes in the storage pool.
- What the next storage pool is, if any of the previous conditions prevent the file from being stored in the storage pool that is being checked.

Using these factors, the server determines if the file can be written to that storage pool or the next storage pool in the hierarchy.

Subfile backups: When the client backs up a subfile, it still reports the size of the entire file. Therefore, allocation requests against server storage and placement in the storage hierarchy are based on the full size of the file. The server does not aggregate a subfile with other files if the size of the entire file is too large to aggregate. For example, the entire file is 8MB, but the subfile is only 10KB. The server does not typically aggregate a large file, so the server begins to store this file as a stand-alone file. However, the client sends only 10KB, and it is now too late for the server to aggregate other files with this 10KB file. As a result, the benefits of aggregation are not always realized when clients back up subfiles.

How the Server Stores Files in a Storage Hierarchy

As an example of how the server stores files in a storage hierarchy, assume a company has a storage pool hierarchy as shown in Figure 17.

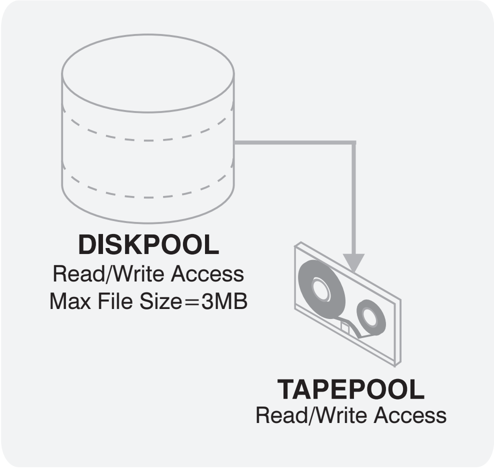


Figure 17. Storage Hierarchy Example

The storage pool hierarchy consists of two storage pools:

DISKPOOL
The top of the storage hierarchy. It contains fast disk volumes for storing data.

TAPEPOOL
The next storage pool in the hierarchy. It contains tape volumes accessed by high-performance tape drives.

Assume a user wants to archive a 5MB file that is named *FileX*. FileX is bound to a management class that contains an archive copy group whose storage destination is DISKPOOL, see Figure 17.

When the user archives the file, the server determines where to store the file based on the following process:

1. The server selects DISKPOOL because it is the storage destination specified in the archive copy group.
2. Because the access mode for DISKPOOL is read/write, the server checks the maximum file size allowed in the storage pool.

   The maximum file size applies to the physical file being stored, which may be a single client file or an aggregate file. The maximum file size allowed in DISKPOOL is 3MB. FileX is a 5MB file and therefore cannot be stored in DISKPOOL.
3. The server searches for the next storage pool in the storage hierarchy.

   If the DISKPOOL storage pool has no maximum file size specified, the server checks for enough space in the pool to store the physical file. If there is not enough space for the physical file, the server uses the next storage pool in the storage hierarchy to store the file.
4. The server checks the access mode of TAPEPOOL, which is the next storage pool in the storage hierarchy. The access mode for TAPEPOOL is read/write.
5. The server then checks the maximum file size allowed in the TAPEPOOL storage pool. Because TAPEPOOL is the last storage pool in the storage

hierarchy, no maximum file size is specified. Therefore, if there is available space in TAPEPOOL, FileX can be stored in it.

Using Copy Storage Pools to Back Up a Storage Hierarchy

Copy storage pools enable you to back up your primary storage pools for an additional level of data protection for clients. See "Backing Up Storage Pools" on page 507 for details. Copy storage pools are not part of a storage hierarchy.

For efficiency, it is recommended that you use one copy storage pool to back up all primary storage pools that are linked to form a storage hierarchy. By backing up all primary storage pools to one copy storage pool, you do not need to recopy a file when the file migrates from its original primary storage pool to another primary storage pool in the storage hierarchy.

When defining copy storage pools to primary pools that have defined next pools, the copy pool list for each primary pool should be the same. Defining different copy pool lists can cause resources to be freed when failing over to the next pool. If the resources are freed, it can delay the completion of client operations.

In most cases, a single copy storage pool can be used for backup of all primary storage pools. The number of copy storage pools you need depends on whether you have more than one primary storage pool hierarchy and on what type of disaster recovery protection you want to implement.

Multiple copy storage pools may be needed to handle particular situations, including:

- Special processing of certain primary storage hierarchies (for example, archive storage pools or storage pools dedicated to priority clients)
- Creation of multiple copies for multiple locations (for example, to keep one copy onsite and one copy offsite)
- Rotation of full storage pool backups (see "Backing Up Storage Pools" on page 507 for more information)

Using the Hierarchy to Stage Client Data from Disk to Tape

A common way to use the storage hierarchy is to initially store client data on disk, then let the server migrate the data to tape. Typically you would need to ensure that you have enough disk storage to handle one night's worth of the clients' incremental backups. While not always possible, this guideline proves to be valuable when considering storage pool backups.

For example, if you have enough disk space for nightly incremental backups for clients and have tape devices, you can set up the following pools:

- A primary storage pool on disk, with enough volumes assigned to contain the nightly incremental backups for clients
- A primary storage pool on tape, which is identified as the next storage pool in the hierarchy for the disk storage pool
- A copy storage pool on tape

You can then schedule the following steps every night:

1. Perform an incremental backup of the clients to the disk storage pool.
2. After clients complete their backups, back up the disk primary storage pool (now containing the incremental backups) to the copy storage pool.

Backing up disk storage pools before migration processing allows you to copy as many files as possible while they are still on disk. This saves mount requests while performing your storage pool backups.

3. Start the migration of the files in the disk primary storage pool to the tape primary storage pool (the next pool in the hierarchy) by lowering the high migration threshold. For example, lower the threshold to 40%.

   When this migration completes, raise the high migration threshold back to 100%.

4. Back up the tape primary storage pool to the copy storage pool to ensure that all files have been backed up.

   The tape primary storage pool must still be backed up to catch any files that might have been missed in the backup of the disk storage pools (for example, large files that went directly to sequential media).

See "Estimating Space Needs for Storage Pools" on page 196 for more information about storage pool space.

Migration of Files in a Storage Pool Hierarchy

The server provides automatic migration to maintain free space in a primary storage pool. The server can migrate data from one storage pool to the next storage pool in the hierarchy. This process helps to ensure that there is sufficient free space in the storage pools at the top of the hierarchy, where faster devices can provide the most benefit to clients. For example, the server can migrate data stored in a random access disk storage pool to a slower but less expensive sequential access storage pool.

Note: You should not run MOVE DATA or MOVE NODEDATA and migration processing concurrently on the same storage pool.

You can control:

When migration begins and ends
You use migration thresholds to control when migration begins and ends. Thresholds are set as levels of the space that is used in a storage pool, expressed as a percent of total space available in the storage pool. For a disk storage pool, the server compares the threshold with a calculation of the amount of data stored in the pool as a percent of the actual data capacity of the volumes in the pool. For a sequential access storage pool, the server compares the threshold with a calculation of the number of volumes containing data as a percent of the total number of volumes available to the pool.

How the server chooses files to migrate
By default, the server does not consider how long a file has been in a storage pool or how long since a file was accessed before choosing files to migrate. Optional parameters allow you to change the default. You can ensure that files remain in a storage pool for a minimum amount of time before the server migrates them to another pool. To do this, you set a migration delay period for a storage pool. Before the server can migrate a file, the file must be stored in the storage pool at least as long as the migration delay period. For disk storage pools, the last time the file was accessed is also considered for migration delay.

Migration processing differs for disk storage pools versus sequential access storage pools. If you plan to modify the default migration parameter settings for storage pools or want to understand how migration works, you should read the following sections:

“Migration for Disk Storage Pools”

“Migration for Sequential Access Storage Pools” on page 180

Migration for Disk Storage Pools

When you define or update a storage pool, you can set migration thresholds to specify when the server should begin and stop migrating data to the next storage pool in the storage hierarchy. Migration thresholds are defined in terms of a percentage of total data capacity for the disk storage pool. You can use the defaults for the migration thresholds, or you can change the threshold values to identify the maximum and minimum amount of space for a storage pool. See “How the Server Selects Files to Migrate” and “Choosing Appropriate Migration Threshold Values” on page 178 for more information about migration thresholds.

You can control how long files must stay in a storage pool before they are eligible for migration by setting a migration delay for a storage pool. See “Keeping Files in a Storage Pool” on page 179.

If you decide to enable cache for disk storage pools, files can temporarily remain on disks even after migration. You may want to set migration thresholds lower when you use cache. See “Minimizing Access Time to Migrated Files” on page 180 and “Using Cache on Disk Storage Pools” on page 183 for information about using the cache.

How the Server Selects Files to Migrate

When data in a storage pool uses a percentage of the pool’s capacity that is equal to the high migration threshold, the server migrates files from the pool to the next storage pool. The server selects the files to migrate as follows:

1. The server checks for the client node that has backed up or migrated the largest single file space or has archived files that occupy the most space.
2. For *all* files from *every* file space belonging to the client node that was identified, the server examines the number of days since the files were stored in the storage pool and last retrieved from the storage pool. The server compares the number (whichever is less) to the migration delay that is set for the storage pool. The server migrates any of these files for which the number is more than the migration delay set for the storage pool.
3. After the server migrates the files for the first client node to the next storage pool, the server checks the low migration threshold for the storage pool. If the amount of space that is used in the storage pool is now below the low migration threshold, migration ends. If not, the server chooses another client node by using the same criteria as described above, and the migration process continues.

The server may not be able to reach the low migration threshold for the pool by migrating only files that have been stored longer than the migration delay period. When this happens, the server checks the storage pool characteristic that determines whether migration should stop even if the pool is still above the low migration threshold. See “Keeping Files in a Storage Pool” on page 179 for more information.

If multiple migration processes are running (controlled by the MIGPROCESS parameter of the DEFINE STGPOOL command), the server may choose the files from more than one node for migration at the same time.

For example, Table 16 displays information that is contained in the database that is used by the server to determine which files to migrate. This example assumes that the storage pool contains no space-managed files. This example also assumes that the migration delay period for the storage pool is set to zero, meaning any files can be migrated regardless of time stored in the pool or the last time of access.

Table 16. Database Information on Files Stored in DISKPOOL

Client Node	Backed-Up File Spaces and Sizes		Archived Files (All Client File Spaces)
TOMC	TOMC/C	200MB	55MB
	TOMC/D	100MB	
CAROL	CAROL	50MB	5MB
PEASE	PEASE/home	150MB	40MB
	PEASE/temp	175MB	

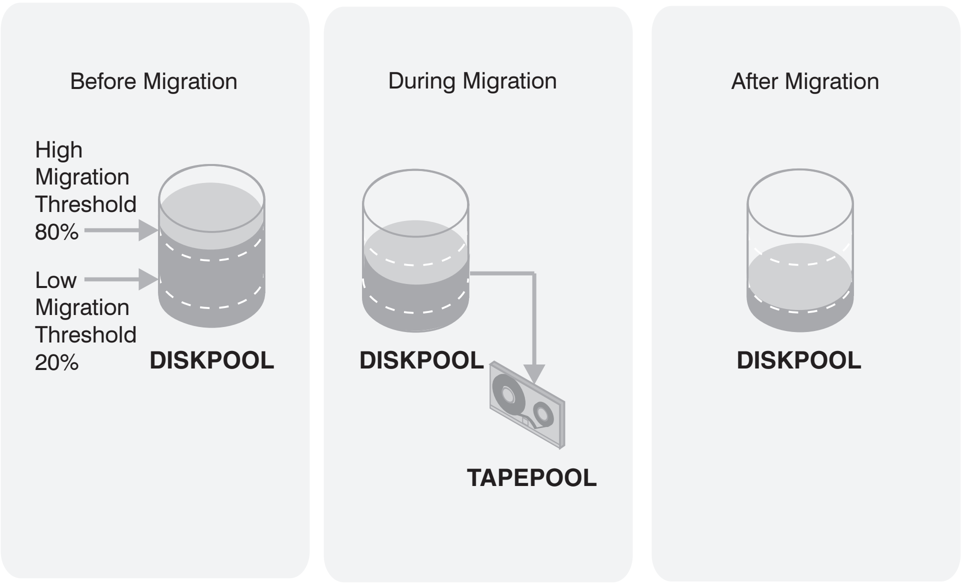


Figure 18. The Migration Process and Migration Thresholds

Figure 18 shows what happens when the high migration threshold defined for the disk storage pool DISKPOOL is exceeded. When the amount of migratable data in DISKPOOL reaches 80%, the server performs the following tasks:

1. Determines that the TOMC/C file space is taking up the most space in the DISKPOOL storage pool, more than any other single backed-up or space-managed file space and more than any client node's archived files.
2. Locates all data belonging to node TOMC stored in DISKPOOL. In this example, node TOMC has backed up or archived files from file spaces TOMC/C and TOMC/D stored in the DISKPOOL storage pool.
3. Migrates all data from TOMC/C and TOMC/D to the next available storage pool. In this example, the data is migrated to the tape storage pool, TAPEPOOL.

The server migrates all of the data from both file spaces belonging to node TOMC, even if the occupancy of the storage pool drops below the low migration threshold before the second file space has been migrated.

If the cache option is enabled, files that are migrated remain on disk storage (that is, the files are *cached*) until space is needed for new files. For more information about using cache, see "Using Cache on Disk Storage Pools" on page 183.

4. After all files that belong to TOMC are migrated to the next storage pool, the server checks the low migration threshold. If the low migration threshold has not been reached, then the server again determines which client node has backed up or migrated the largest single file space or has archived files that occupy the most space. The server begins migrating files belonging to that node.

   In this example, the server migrates *all* files that belong to the client node named PEASE to the TAPEPOOL storage pool.
5. After all the files that belong to PEASE are migrated to the next storage pool, the server checks the low migration threshold again. If the low migration threshold has been reached or passed, then migration ends.

Choosing Appropriate Migration Threshold Values

Setting migration thresholds for disk storage pools ensures sufficient free space on faster speed devices, which can lead to better performance. Choosing thresholds appropriate for your situation takes some experimenting, and you can start by using the default values. You need to ensure that migration occurs frequently enough to maintain some free space but not so frequently that the device is unavailable for other use.

Choosing the High-Migration Threshold: To choose the high-migration threshold, consider:

- The amount of storage capacity provided for each storage pool
- The amount of free storage needed for users to store additional files, without having migration occur

If you set the high-migration threshold too high, the pool may be just under the high threshold, but not have enough space to store an additional, typical client file. Or, with a high threshold of 100%, the pool may become full and a migration process must start before clients can back up any additional data to the disk storage pool. In either case, the server stores client files directly to tape until migration completes, resulting in slower performance.

If you set the high-migration threshold too low, migration runs more frequently and can interfere with other operations.

Keeping the high-migration threshold at a single value means that migration processing could start at any time of day, whenever that threshold is exceeded. You can control when migration occurs by using administrative command schedules to change the threshold. For example, set the high-migration threshold to 95% during the night when clients run their backup operations. Lower the high-migration threshold to 50% during the time of day when you want migration to occur. By scheduling when migration occurs, you can choose a time when your tape drives and mount operators are available for the operation.

Choosing the Low-Migration Threshold: To choose the low-migration threshold, consider:

- The amount of free disk storage space needed for normal daily processing. If you have disk space to spare, you can keep more data on the disk (a larger low threshold). If clients' daily backups are enough to fill the disk space every day, you may need to empty the disk (a smaller low threshold).

  If your disk space is limited, try setting the threshold so that migration frees enough space for the pool to handle the amount of client data that is typically stored every day. Migration then runs about every day, or you can force it to run every day by lowering the high-migration threshold at a time you choose.

  You may also want to identify clients that are transferring large amounts of data daily. For these clients, you may want to set up policy (a new copy group or a new policy domain) so that their data is stored directly to tape. Using a separate policy in this way can optimize the use of disk for the majority of clients.
- Whether you use cache on disk storage pools to improve how quickly some files are retrieved. If you use cache, you can set the low threshold lower, yet still maintain faster retrieval for some data. Migrated data remains cached on the disk until new client data pushes the data off the disk. Using cache requires more disk space for the database, however, and can slow backup and archive operations that use the storage pool.

  If you do not use cache, you may want to keep the low threshold at a higher number so that more data stays on the disk.
- How frequently you want migration to occur, based on the availability of sequential access storage devices and mount operators. The larger the low threshold, the shorter time that a migration process runs (because there is less data to migrate). But if the pool refills quickly, then migration occurs more frequently. The smaller the low threshold, the longer time that a migration process runs, but the process runs less frequently.

  You may need to balance the costs of larger disk storage pools with the costs of running migration (drives, tapes, and either operators or automated libraries).
- Whether you are using collocation on the next storage pool. When you use collocation, the server attempts to store data for different clients or client file spaces on separate tapes, even for clients with small amounts of data. You may want to set the low threshold to keep more data on disk, to avoid having many tapes used by clients with only small amounts of data.

Keeping Files in a Storage Pool

For some applications, you may want to ensure that files remain in the storage pool where they were initially stored by the server for a certain period of time. For example, you may have backups of monthly summary data that you want to keep in your disk storage pool for quicker access until the data is 30 days old. After the 30 days, the server can then move the file off into a tape storage pool.

You can delay migration of files for a specified number of days. The number of days is counted from the day that a file was stored in the storage pool or retrieved by a client, whichever is more recent. You can set the migration delay separately for each storage pool. When you set the delay to zero, the server can migrate any file from the storage pool, regardless of how short a time the file has been in the storage pool. When you set the delay to greater than zero, the server checks whether the file has been in the storage pool for at least the migration delay period before migrating the file.

Note: If you want the number of days for migration delay to be counted based only on when a file was stored and not when it was retrieved, use the NORETRIEVEDATE server option. See *Administrator's Reference* for more information on the server option.

If you set migration delay for a pool, you need to decide what is more important: either ensuring that files stay in the storage pool for the migration delay period, or ensuring that there is enough space in the storage pool for new files. For each storage pool that has a migration delay set, you can choose what happens as the server tries to move enough data out of the storage pool to reach the low migration threshold. If the server cannot reach the low migration threshold by moving only files that have been stored longer than the migration delay, you can choose one of the following:

- Allow the server to move files out of the storage pool even if they have not been in the pool for the migration delay (MIGCONTINUE=YES). This is the default. Allowing migration to continue ensures that space is made available in the storage pool for new files that need to be stored there.
- Have the server stop migration without reaching the low migration threshold (MIGCONTINUE=NO). Stopping migration ensures that files remain in the storage pool for the time you specified with the migration delay. The administrator must ensure that there is always enough space available in the storage pool to hold the data for the required number of days.

If you allow more than one migration process for the storage pool and allow the server to move files that do not satisfy the migration delay time (MIGCONTINUE=YES), some files that do not satisfy the migration delay time may be migrated unnecessarily. As one process migrates files that satisfy the migration delay time, a second process could begin migrating files that do not satisfy the migration delay time to meet the low migration threshold. The first process that is still migrating files that satisfy the migration delay time might have, by itself, caused the storage pool to meet the low migration threshold.

Minimizing Access Time to Migrated Files

Caching is a method of minimizing access time to files on disk storage, even if the server has migrated files to a tape storage pool. However, cached files are removed from disk when the space they occupy is required. The file then must be obtained from the storage pool to which it was migrated.

Note: The use of cache has some disadvantages. See "Using Cache on Disk Storage Pools" on page 183.

To ensure that files remain on disk storage and do not migrate to other storage pools, use one of the following methods:

- Do not define the *next* storage pool.

  A disadvantage of using this method is that if the file exceeds the space available in the storage pool, the operation to store the file fails.
- Set the high-migration threshold to 100%.

  When you set the high migration threshold to 100%, files will not migrate at all. You can still define the *next* storage pool in the storage hierarchy, and set the maximum file size so that large files are stored in the next storage pool in the hierarchy.

  A disadvantage of setting the high threshold to 100% is that once the pool becomes full, client files are stored directly to tape instead of to disk. Performance may be affected as a result.

Migration for Sequential Access Storage Pools

You can set up migration thresholds for sequential access storage pools. However, you probably will not want the server to perform this type of migration on a regular basis. An operation such as tape-to-tape migration has limited benefits compared to disk-to-tape migration, and requires at least two tape drives. Migrating data from one sequential access storage pool to another may be appropriate in some cases, for example, when you install a tape drive that uses a different type of tape and want to move data to that tape.

To control the migration process, you can set migration thresholds and a migration delay for each storage pool.

Note:

- You can migrate data from a sequential access storage pool only to another sequential access storage pool. You cannot migrate data from a sequential access storage pool to a disk storage pool. If you need to move data from a sequential access storage pool to a disk storage pool, use the MOVE DATA command. See “Moving Files from One Volume to Another Volume” on page 211.
- Storage pools using the NETAPPDUMP data format are unable to use migration.

How Tivoli Storage Manager Migrates Data from Sequential Access Storage Pools

The server begins the migration process when the number of volumes containing data as a percentage of the total volumes in the storage pool reaches the high migration threshold. The server migrates data from sequential storage pools by volume, to minimize the number of mounts for volumes. The server performs the following processing for migration:

1. The server first reclaims volumes that have exceeded the reclamation threshold. Reclamation is a server process of consolidating data from several volumes onto one volume. (See “Reclaiming Space in Sequential Access Storage Pools” on page 189.)
2. After reclamation processing, the server compares the space used in the storage pool to the low migration threshold.
3. If the space used is now below the low migration threshold, the server stops processing. If the space used is still above the low migration threshold, the server determines which volume is the least recently referenced volume.
4. If the number of days since data was written is greater than the migration delay, the server migrates the volume. Otherwise, the server does not migrate this volume.
5. The server repeats steps 3 and 4 until the storage pool reaches the low migration threshold.

Because migration delay can prevent volumes from being migrated, the server can migrate data from all eligible volumes yet still find that the storage pool is above the low migration threshold. If you set migration delay for a pool, you need to decide what is more important: either ensuring that data stays in the storage pool for as long as the migration delay, or ensuring there is enough space in the storage pool for new data. For each storage pool that has a migration delay set, you can choose what happens as the server tries to move enough data out of the storage pool to reach the low migration threshold. If the server cannot reach the low migration threshold by migrating only volumes that meet the migration delay requirement, you can choose one of the following:

- Allow the server to migrate volumes from the storage pool even if they do not meet the migration delay criteria (MIGCONTINUE=YES). This is the default. Allowing migration to continue ensures that space is made available in the storage pool for new files that need to be stored there.

- Have the server stop migration without reaching the low migration threshold (MIGCONTINUE=NO). Stopping migration ensures that volumes are not migrated for the time you specified with the migration delay. The administrator must ensure that there is always enough space available in the storage pool to hold the data for the required number of days.

Selecting Migration Criteria for Sequential Access Storage Pools

When defining migration criteria for sequential access storage pools, consider:

- The capacity of the volumes in the storage pool
- The time required to migrate data to the next storage pool
- The speed of the devices that the storage pool uses
- The time required to mount media, such as tape volumes, into drives
- Whether operator presence is required

If you decide to migrate data from one sequential access storage pool to another, ensure that:

- Two drives (mount points) are available, one in each storage pool.
- The access mode for the next storage pool in the storage hierarchy is set to read/write.

  For information about setting an access mode for sequential access storage pools, see "Defining or Updating Primary Storage Pools" on page 158.
- Collocation is set the same in both storage pools. For example, if collocation is set to *yes* in the first storage pool, then collocation should be set to *yes* in the next storage pool.

  When you enable collocation for a storage pool, the server attempts to keep all files belonging to a client node or a client file space on a minimal number of volumes. For information about collocation for sequential access storage pools, see "Keeping a Client's Files Together: Collocation" on page 184.
- You have sufficient staff available to handle any necessary media mount and dismount operations. More mount operations occur because the server attempts to reclaim space from sequential access storage pool volumes before it migrates files to the next storage pool.

  If you want to limit migration from a sequential access storage pool to another storage pool, set the high-migration threshold to a high percentage, such as 95%.

  For information about setting a reclamation threshold for tape storage pools, see "Reclaiming Space in Sequential Access Storage Pools" on page 189.

There is no straightforward way to selectively migrate data for a specific node from one sequential storage pool to another. If you know the volumes on which a particular node's data is stored, you can use the MOVE DATA command to move all files from selected volumes to the new storage pool. See "Moving Files from One Volume to Another Volume" on page 211.

Migration and Copy Storage Pools

Copy storage pools are not part of the hierarchy for migration. Files are not migrated to or from copy storage pools. The only way to store files in copy storage pools is by backing up primary storage pools (the BACKUP STGPOOL command).

Migration of files between primary storage pools does not affect copy storage pool files. Copy storage pool files do not move when primary storage pool files move.

For example, suppose a copy of a file is made while it is in a disk storage pool. The file then migrates to a primary tape storage pool. If you then back up the primary tape storage pool to the same copy storage pool, a new copy of the file is not needed. The server knows it already has a valid copy of the file.

Using Cache on Disk Storage Pools

When defining or updating disk storage pools, you can enable or disable cache.

When cache is disabled and migration occurs, the server migrates the files to the next storage pool and erases the files from the disk storage pool. By default, the system disables caching for each disk storage pool because of the potential effects of cache on backup performance.

You can enable cache by specifying CACHE=YES when you define or update a storage pool. When cache is enabled, the migration process leaves behind duplicate copies of files on disk after the server migrates these files to the next storage pool in the storage hierarchy. The copies remain in the disk storage pool, but in a *cached* state, so that subsequent retrieval requests can be satisfied quickly. However, if space is needed to store new data in the disk storage pool, cached files are erased and the space they occupied is used for the new data.

The advantage of using cache for a disk storage pool is that cache can improve how quickly the server retrieves some files. When you use cache, a copy of the file remains on disk storage after the server migrates the primary file to another storage pool. You may want to consider using a disk storage pool with cache enabled for storing space-managed files that are frequently accessed by clients.

However, using cache has some important disadvantages:

- Using cache can increase the time required for client backup operations to complete. Performance is affected because, as part of the backup operation, the server must erase cached files to make room for storing new files. The effect can be severe when the server is storing a very large file and must erase cached files.

  For the best performance for client backup operations to disk storage pools, do not use cache.
- Using cache can require more space for the server database. When you use cache, more database space is needed because the server has to keep track of both the cached copy of the file and the new copy in the next storage pool.

If you leave cache disabled, you may want to consider higher migration thresholds for the disk storage pool. A higher migration threshold keeps files on disk longer because migration occurs less frequently.

How the Server Removes Cached Files

When space is needed, the server reclaims space occupied by cached files. Files that have the oldest retrieval date and occupy the largest amount of disk space are overwritten first. For example, assume that two files, File A and File B, are cached files that are the same size. If File A was last retrieved on 05/16/99 and File B was last retrieved on 06/19/99, then File A is deleted to reclaim space first.

You can change whether the server tracks the retrieval date for a file with the server option, NORETRIEVEDATE. When you include this option in the server options file, the server does not update the retrieval date for files. As a result, the server may remove copies of files in cache even though clients retrieved the files recently.

Effect of Caching on Storage Pool Statistics

The space utilization statistic for the pool (Pct Util) includes the space used by any cached copies of files in the storage pool. The migratable data statistic (Pct Migr) does *not* include space occupied by cached copies of files. The server uses the statistic on migratable data (Pct Migr) to compare with migration threshold parameters to determine when migration should begin or end. For more information on storage pool statistics, see "Monitoring Storage Pools and Volumes" on page 198.

Keeping a Client's Files Together: Collocation

With *collocation* enabled, the server attempts to keep files belonging to a single client node or to a single file space of a client node on a minimal number of sequential access storage volumes. You can set collocation for each sequential access storage pool when you define or update the pool.

To have the server collocate data in a storage pool by client node, set collocation to YES. To have the server collocate data in a storage pool by client file space, set collocation to FILESPACE. By using collocation, you reduce the number of volume mount operations required when users restore, retrieve, or recall many files from the storage pool. Collocation thus improves access time for these operations. Figure 19 shows an example of collocation by client node with three clients, each having a separate volume containing that client's data.

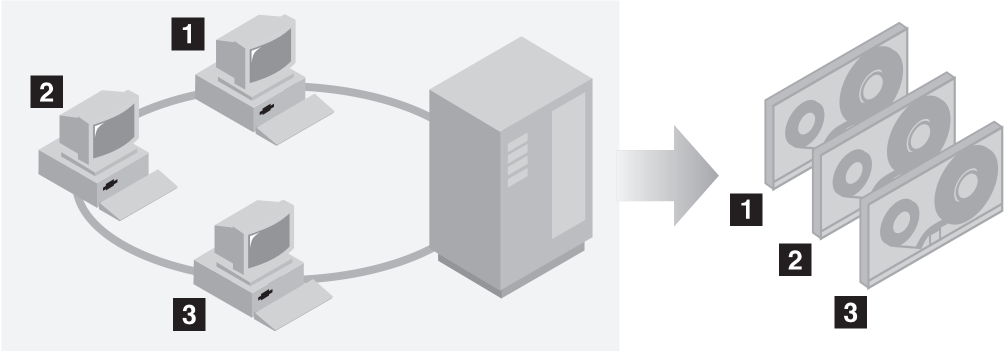


Figure 19. Example of Collocation Enabled

When collocation is disabled, the server attempts to use all available space on each volume before selecting a new volume. While this process provides better utilization of individual volumes, user files can become scattered across many volumes. Figure 20 on page 185 shows an example of collocation disabled, with three clients sharing space on a volume.

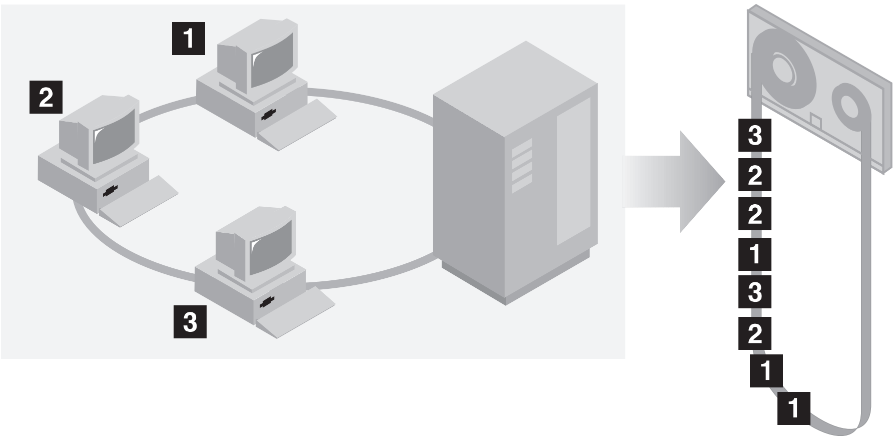


Figure 20. Example of Collocation Disabled

With collocation disabled, when users restore, retrieve, or recall a large number of files, media mount operators may be required to mount more volumes. The system default is to not use collocation.

The following sections give more detail on collocation:

“The Effects of Collocation on Operations”

“How the Server Selects Volumes with Collocation Enabled” on page 186

“How the Server Selects Volumes with Collocation Disabled” on page 187

“Turning Collocation On or Off” on page 188

“Collocation on Copy Storage Pools” on page 188

The Effects of Collocation on Operations

Table 17 summarizes the effects of collocation on operations.

Table 17. Effect of Collocation on Operations

Operation	Collocation Enabled	Collocation Disabled
Backing up, archiving, or migrating client files	More media mounts to collocate files.	Usually fewer media mounts are required.
Restoring, retrieving or recalling client files	Large numbers of files can be restored, retrieved, or recalled more quickly because files are located on fewer volumes.	Multiple mounts of media may be required for a single user because files may be spread across multiple volumes. More than one user’s files can be stored on the same sequential access storage volume. For example, if two users attempt to recover a file that resides on the same volume, the second user will be forced to wait until the first user’s files are recovered.
Storing data on tape	The server attempts to use all available tape volumes to separate user files before it uses all available space on every tape volume.	The server attempts to use all available space on each tape volume before using another tape volume.

Table 17. Effect of Collocation on Operations (continued)

Operation	Collocation Enabled	Collocation Disabled
Media mount operations	More mount operations when user files are backed up, archived, or migrated from client nodes directly to sequential access volumes. More mount operations during reclamation and storage pool migration. More volumes to handle because volumes are not fully used.	More mount operations required during restore, retrieve, and recall of client files.

Tip: If you use collocation, but want to reduce the number of media mounts and use space on sequential volumes more efficiently, you can do the following:

- Define a storage pool hierarchy and policy to require that backed-up, archived, or space-managed files are stored initially in disk storage pools.

  When files are migrated from a disk storage pool, the server attempts to migrate all files belonging to the client node that is using the most disk space in the storage pool. This process works well with the collocation option because the server tries to place all of the files from a given client on the same sequential access storage volume.
- Use scratch volumes for sequential access storage pools to allow the server to select new volumes for collocation.

How the Server Selects Volumes with Collocation Enabled

When collocation at the client node level is enabled for a storage pool (COLLOCATION=YES) and a client node backs up, archives, or migrates files to the storage pool, the server attempts to select a volume using the following selection order:

1. A volume that already contains files from the same client node
2. An empty predefined volume
3. An empty scratch volume
4. A volume with the most available free space among volumes that already contain data

When collocation at the file space level is enabled for a storage pool (COLLOCATION=FILESPACE) and a client node backs up, archives, or migrates files to the storage pool, the server attempts to select a volume using the following selection order:

1. A volume that already contains files from the same file space of that client node
2. An empty predefined volume
3. An empty scratch volume
4. A volume containing data from the same client node
5. A volume with the most available free space among volumes that already contain data

When the server needs to continue to store data on a second volume, it uses the following selection order to acquire additional space:

1. An empty predefined volume

2. An empty scratch volume
3. A volume with the most available free space among volumes that already contain data
4. Any available volume in the storage pool

Through this selection process, the server attempts to provide the best use of individual volumes while minimizing the mixing of files from different clients or file spaces on volumes. For example, Figure 21 shows that volume selection is *horizontal*, where all available volumes are used before all available space on each volume is used. A, B, C, and D represent files from four different client nodes.

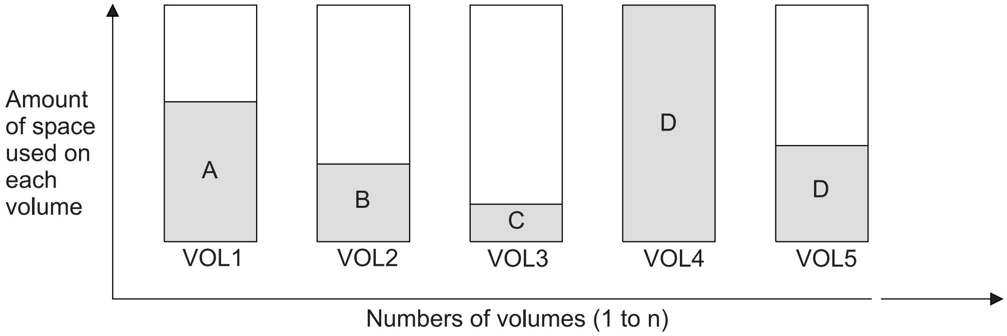

Amount of space used on each volume
A
B
C
D
D
VOL1
VOL2
VOL3
VOL4
VOL5
Numbers of volumes (1 to n)

Figure 21. Using All Available Sequential Access Storage Volumes with Collocation Enabled

How the Server Selects Volumes with Collocation Disabled

When collocation is disabled, the server attempts to use all available space in a storage volume before it accesses another volume. When storing client files in a sequential access storage pool where collocation is disabled, the server selects a volume using the following selection order:

1. A previously used sequential volume with available space (a volume with the most amount of data is selected first)
2. An empty volume

When the server needs to continue to store data on a second volume, it attempts to select an empty volume. If none exists, the server attempts to select any remaining available volume in the storage pool.

Figure 22 on page 188 shows that volume utilization is *vertical* when collocation is disabled. In this example, fewer volumes are used because the server attempts to use all available space by mixing client files on individual volumes. A, B, C, and D represent files from four different client nodes.

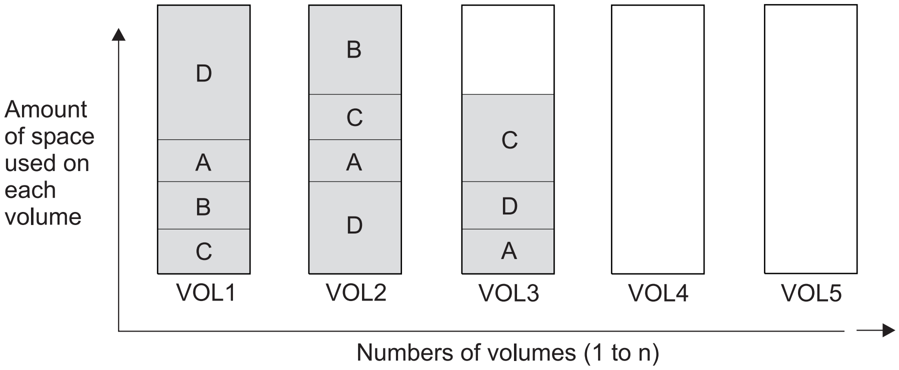

Amount of space used on each volume
D
A
B
C
VOL1
B
C
A
D
VOL2
C
D
A
VOL3
VOL4
VOL5
Numbers of volumes (1 to n)

Figure 22. Using All Available Space on Sequential Volumes with Collocation Disabled

Turning Collocation On or Off

After you define a storage pool, you can change the collocation setting by updating the storage pool. The change in collocation for the pool does not affect files that are already stored in the pool.

For example, if collocation had been off for a storage pool and you turn it on, *from then on* client files stored in the pool are collocated. Files that had previously been stored in the pool are *not* moved to collocate them. As volumes are reclaimed, however, the data in the pool tends to become more collocated. You can also use the MOVE DATA command to move data to new volumes to increase collocation, if you are able to afford the processing time and the volume mount activity this would cause.

Collocation on Copy Storage Pools

Using collocation on copy storage pools requires special consideration.

Primary and copy storage pools perform different recovery roles. Normally you use primary storage pools to recover data to clients directly. You use copy storage pools to recover data to the primary storage pools. In a disaster where both clients and the server are lost, the copy storage pool volumes will probably be used directly to recover clients. The types of recovery scenarios that concern you the most will help you to determine whether to use collocation on your copy storage pools.

You may also want to consider that collocation on copy storage pools will result in more partially filled volumes and potentially unnecessary offsite reclamation activity. Collocation typically results in a partially filled sequential volume for each client or client file space. This may be acceptable for primary storage pools because these partially filled volumes remain available and can be filled during the next migration process. However, for copy storage pools this may be unacceptable because the storage pool backups are usually made to be taken offsite immediately. If you use collocation for copy storage pools, you will have to decide between:

- Taking more partially filled volumes offsite, thereby increasing the reclamation activity when the reclamation threshold is lowered or reached.

or

- Leaving these partially filled volumes onsite until they fill and risk not having an offsite copy of the data on these volumes.

With collocation disabled for a copy storage pool, typically there will be only a few partially filled volumes after storage pool backups to the copy storage pool are complete.

Consider carefully before using collocation for copy storage pools. Even if you use collocation for your primary storage pools, you may want to disable collocation for copy storage pools. Collocation on copy storage pools may be desirable when you have few clients, but each of them has large amounts of incremental backup data each day.

See “Keeping a Client’s Files Together: Collocation” on page 184 for more information about collocation.

Reclaiming Space in Sequential Access Storage Pools

Space on a sequential volume becomes reclaimable as files expire or are deleted from the volume. For example, files become obsolete because of aging or limits on the number of versions of a file. In reclamation processing, the server rewrites files on the volume being reclaimed to other volumes in the storage pool, making the reclaimed volume available for reuse.

The server reclaims the space in storage pools based on a *reclamation threshold* that you can set for each sequential access storage pool. When the percentage of space that can be reclaimed on a volume rises above the reclamation threshold, the server reclaims the volume. See the following sections:

“How Tivoli Storage Manager Reclamation Works”
“Choosing a Reclamation Threshold” on page 191
“Reclaiming Volumes in a Storage Pool with One Drive” on page 192
“Reclamation for WORM Optical Media” on page 192
“Reclamation of Volumes with the Device Type of SERVER” on page 192
“Reclamation for Copy Storage Pools” on page 193
“How Collocation Affects Reclamation” on page 195

Note: Storage pools using the NETAPPDUMP data format are unable to use reclamation.

How Tivoli Storage Manager Reclamation Works

When the percentage of reclaimable space on a volume exceeds the reclamation threshold set for the storage pool, the volume is eligible for reclamation. The server checks whether reclamation is needed at least once per hour and begins space reclamation for eligible volumes. You can set a reclamation threshold for each sequential access storage pool when you define or update the pool.

During space reclamation, the server copies files that remain on eligible volumes to other volumes. For example, Figure 23 on page 190 shows that the server consolidates the files from tapes 1, 2, and 3 on tape 4. During reclamation, the server copies the files to volumes in the same storage pool unless you have specified a reclamation storage pool. Use a reclamation storage pool to allow automatic reclamation for a storage pool with only one drive.

The server also reclaims space within an aggregate file. An aggregate is a physical file that contains multiple logical files that are backed up or archived from a client in a single transaction. Space within the file becomes reclaimable space as logical files in the aggregate expire or are deleted by the client. The server removes unused space from expired or deleted logical files as the server copies the aggregate file to another volume during reclamation processing. However, reclamation does not aggregate files that were originally stored in non-aggregated form. Reclamation also does not combine aggregates to make new aggregates. You can also reclaim space in an aggregate by issuing the MOVE DATA command. See "Reclaiming Space in Aggregates During Data Movement" on page 214 for details.

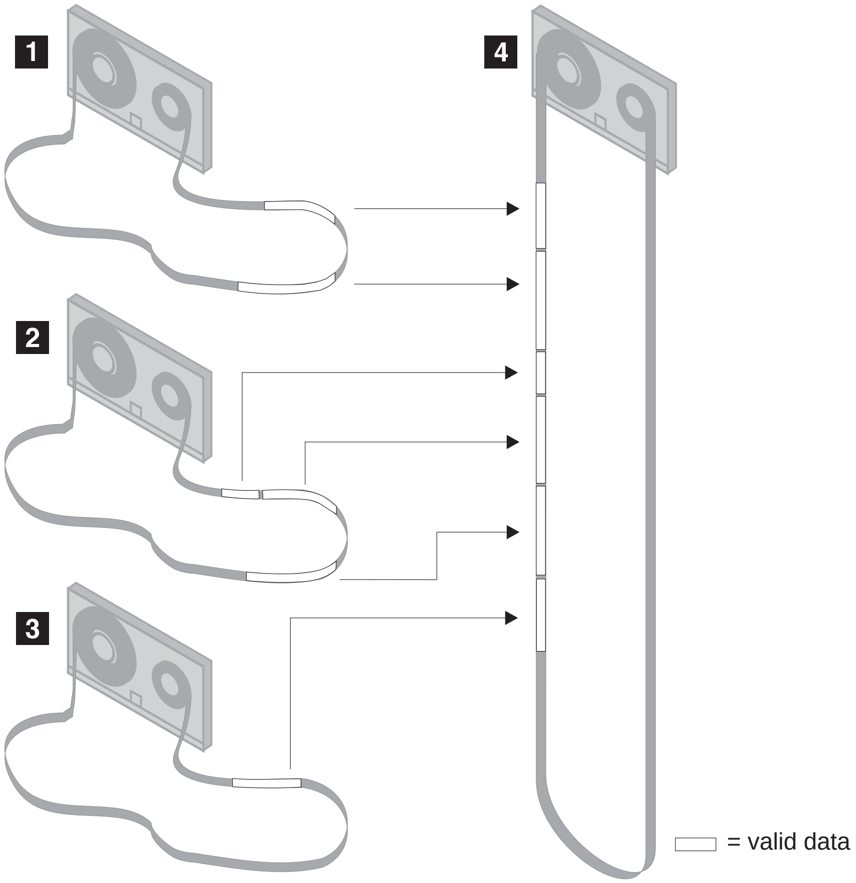


Figure 23. Tape Reclamation

After the server moves all readable files to other volumes, one of the following occurs for the reclaimed volume:

- If you have explicitly defined the volume to the storage pool, the volume becomes available for reuse by that storage pool
- If the server acquired the volume as a scratch volume, the server deletes the volume from the TSM database

Volumes that have a device type of SERVER are reclaimed in the same way as other sequential access volumes. However, because the volumes are actually data stored in the storage of another TSM server, the reclamation process can consume network resources. See "Reclamation of Volumes with the Device Type of SERVER" on page 192 for details of how the server reclaims these types of volumes.

Volumes in a copy storage pool are reclaimed in the same manner as a primary storage pool except for the following:

- *Offsite* volumes are handled differently.

- The server copies active files from the candidate volume only to other volumes in the *same* storage pool.

See "Reclamation for Copy Storage Pools" on page 193 for details.

Choosing a Reclamation Threshold

The reclamation threshold indicates how much reclaimable space a volume must have before the server reclaims the volume. Space is reclaimable because it is occupied by files that have been expired or deleted from the TSM database, or because the space has never been used.

The server checks whether reclamation is needed at least once per hour. The lower the reclamation threshold, the more frequently the server tries to reclaim space. Frequent reclamation optimizes the use of a sequential access storage pool's space, but can interfere with other processes, such as backups from clients.

If the reclamation threshold is high, reclamation occurs less frequently. A high reclamation threshold is useful if mounting a volume is a manual operation and the operations staff is at a minimum.

Each reclamation process requires at least two simultaneous volume mounts, that is, at least two mount points (drives). The two drives must be in the same device class to allow the server to move the data from reclaimed volumes to other volumes in the same storage pool. A sufficient number of volumes, drives (if appropriate), and mount operators (if appropriate) must be available to handle frequent reclamation requests. For more information about mount limit, see "Mount Limit" on page 143. If the device class for the storage pool does not have two drives, you can specify a reclamation storage pool. For information about how to use a reclamation storage pool for storage pools with only one mount point, see "Reclaiming Volumes in a Storage Pool with One Drive" on page 192.

If you set the reclamation threshold to 50% or greater, the server can combine the usable files from two or more volumes onto a single new volume.

Setting the reclamation threshold to 100% prevents reclamation from occurring. You might want to do this to control when reclamation occurs, to prevent interfering with other server processes. When it is convenient for you and your users, you can lower the reclamation threshold to cause reclamation to begin.

Lowering the Migration Threshold

If you have been running with a high migration threshold and decide you now need to reclaim volumes, you may want to lower the threshold in several steps. Lowering the threshold in steps ensures that volumes with the most reclaimable space are reclaimed first. For example, if you had set the high migration threshold to 100%, first lower the threshold to 98%. Volumes that have reclaimable space of 98% or greater are reclaimed by the server. Lower the threshold again to reclaim more volumes.

If you lower the reclamation threshold while a reclamation process is active, the reclamation process does not immediately stop. If an onsite volume is being reclaimed, the server uses the new threshold setting when the process begins to reclaim the next volume. If offsite volumes are being reclaimed, the server does not use the new threshold setting during the process that is running (because all eligible offsite volumes are reclaimed at the same time).

Use the CANCEL PROCESS command to stop a reclamation process.

Reclaiming Volumes in a Storage Pool with One Drive

When a storage pool has only one mount point (that is, just one drive) available to it through the device class, data cannot be reclaimed from one volume to another within that same storage pool.

To enable volume reclamation for a storage pool that has only one mount point, you can define a *reclamation storage pool* for the server to use when reclaiming volumes. When the server reclaims volumes, the server moves the data from volumes in the original storage pool to volumes in the reclamation storage pool. The server always uses the reclamation storage pool when one is defined, even when the mount limit is greater than one.

If the reclamation storage pool does not have enough space to hold all of the data being reclaimed, the server moves as much of the data as possible into the reclamation storage pool. Any data that could not be moved to volumes in the reclamation storage pool still remains on volumes in the original storage pool.

The pool identified as the reclamation storage pool must be a primary sequential storage pool. The primary purpose of the reclamation storage pool is for temporary storage of reclaimed data. To ensure that data moved to the reclamation storage pool eventually moves back into the original storage pool, specify the original storage pool as the next pool in the storage hierarchy for the reclamation storage pool. For example, if you have a tape library with one drive, you can define a storage pool to be used for reclamation using a device class with a device type of FILE:

```
define stgpool reclaimpool fileclass maxscratch=100
```

Define the storage pool for the tape drive as follows:

```
define stgpool tapepool1 tapeclass maxscratch=100
reclaimstgpool=reclaimpool
```

Finally, update the reclamation storage pool so that data migrates back to the tape storage pool:

```
update stgpool reclaimpool nextstgpool=tapepool1
```

Reclamation for WORM Optical Media

Reclamation for WORM volumes does not mean that you can reuse this write-once media. However, reclamation for WORM volumes does allow you to free library space. Reclamation consolidates data from almost empty volumes to other volumes. You can then eject the empty, used WORM volumes and add new volumes.

Storage pools that are assigned to device classes with a device type of WORM, WORM12, or WORM14 have a default reclamation value of 100. This prevents reclamation of WORM optical media. To allow reclamation, you can set the reclamation value to something lower when defining or updating the storage pool.

Reclamation of Volumes with the Device Type of SERVER

When virtual volumes (volumes with the device type of SERVER) in a primary storage pool are reclaimed, the client data stored on those volumes is sent across the network between the source server and the target server. As a result, the reclamation process can tie up your network resources. To control when reclamation starts for these volumes, consider setting the reclamation threshold to

100% for any primary storage pool that uses virtual volumes. Lower the reclamation threshold at a time when your network is less busy, so that the server can reclaim volumes.

For virtual volumes in a copy storage pool, the server reclaims a volume as follows:

1. The source server determines which files on the volume are still valid.
2. The source server obtains these valid files from a primary storage pool, or if necessary, from an onsite volume (not a virtual volume) in another copy storage pool.
3. The source server writes the files to one or more new virtual volumes in the copy storage pool and updates its database.
4. The server issues a message indicating that the volume was reclaimed.

For information about using the SERVER device type, see "Using Virtual Volumes to Store Data on Another Server" on page 469.

Reclamation for Copy Storage Pools

Reclamation of primary storage pool volumes does not affect copy storage pool files.

Reclamation of volumes in copy storage pools is similar to that of primary storage pools. However, most volumes in copy storage pools may be set to an access mode of offsite, making them ineligible to be mounted. When reclamation occurs and how reclamation processing is done depends on whether the volumes are marked as offsite.

For volumes that are not offsite, reclamation usually occurs after the volume is full and then begins to empty because of file deletion. When the percentage of reclaimable space on a volume rises above the reclamation threshold, the server reclaims the volume. Active files on the volume are rewritten to other volumes in the storage pool, making the original volume available for new files.

For offsite volumes, reclamation can occur when the percentage of unused space on the volume is greater than the reclaim parameter value. The unused space includes both space that has never been used on the volume and space that has become empty because of file deletion. During reclamation, the server copies valid files on offsite volumes from the original files in the primary storage pools. In this way, the server copies valid files on offsite volumes without having to mount these volumes. For more information, see "Reclamation of Offsite Volumes".

Reclamation of copy storage pool volumes should be done periodically to allow reuse of partially filled volumes that are offsite. Reclamation can be done automatically by setting the reclamation threshold for the copy storage pool to less than 100%. However, you need to consider controlling when reclamation occurs because of how offsite volumes are treated. For more information, see "Controlling When Reclamation Occurs for Offsite Volumes" on page 194.

Virtual Volumes: Virtual volumes (volumes that are stored on another TSM server through the use of a device type of SERVER) cannot be set to the offsite access mode.

Reclamation of Offsite Volumes

As for volumes with other access values, volumes with the access value of offsite are eligible for reclamation if the amount of empty space on a volume exceeds the reclamation threshold for the copy storage pool. The default reclamation threshold for copy storage pools is 100%, which means that reclamation is not performed.

When an offsite volume is reclaimed, the files on the volume are rewritten to a *read/write* volume. Effectively, these files are moved back to the onsite location. The files may be obtained from the offsite volume after a disaster, if the volume has not been reused and the database backup that you use for recovery references the files on the offsite volume.

The server reclaims an offsite volume as follows:

1. The server determines which files on the volume are still valid.
2. The server obtains these valid files from a primary storage pool, or if necessary, from an onsite volume of a copy storage pool.
3. The server writes the files to one or more volumes in the copy storage pool and updates the database. If a file is an aggregate file with unused space, the unused space is removed during this process.
4. A message is issued indicating that the offsite volume was reclaimed.

   For a single storage pool, the server reclaims all offsite volumes that are eligible for reclamation at the same time. Reclaiming all the eligible volumes at the same time minimizes the tape mounts for primary storage pool volumes.

If you have the Tivoli Disaster Recovery Manager product, see “Moving Backup Volumes Onsite” on page 561.

Controlling When Reclamation Occurs for Offsite Volumes

Suppose you plan to make daily storage pool backups to a copy storage pool, then mark all new volumes in the copy storage pool as *offsite* and send them to the offsite storage location. This strategy works well with one consideration if you are using automatic reclamation (the reclamation threshold is less than 100%).

Each day’s storage pool backups will create a number of new copy storage pool volumes, the last one being only partially filled. If the percentage of empty space on this partially filled volume is higher than the reclaim percentage, this volume becomes eligible for reclamation as soon as you mark it offsite. The reclamation process would cause a new volume to be created with the same files on it. The volume you take offsite would then be empty according to the TSM database. If you do not recognize what is happening, you could perpetuate this process by marking the new partially filled volume offsite.

One way to resolve this situation is to keep partially filled volumes onsite until they fill up. However, this would mean a small amount of your data would be without an offsite copy for another day.

If you send copy storage pool volumes offsite, it is recommended you control copy storage pool reclamation by using the default value of 100. This turns reclamation off for the copy storage pool. You can start reclamation processing at desired times by changing the reclamation threshold for the storage pool. To monitor offsite volume utilization and help you decide what reclamation threshold to use, enter the following command:

```
query volume * access=offsite format=detailed
```

Depending on your data expiration patterns, you may not need to do reclamation of offsite volumes each day. You may choose to perform offsite reclamation on a less frequent basis. For example, suppose you ship copy storage pool volumes to and from your offsite storage location once a week. You can run reclamation for the copy storage pool weekly, so that as offsite volumes become empty they are sent back for reuse.

When you do perform reclamation for offsite volumes, the following sequence is recommended:

1. Back up your primary storage pools to copy storage pools.
2. Turn on reclamation for copy storage pools by lowering the reclamation threshold below 100%.
3. When reclamation processing completes, turn off reclamation for copy storage pools by raising the reclamation threshold to 100%.
4. Mark any newly created copy storage pool volumes as offsite and then move them to the offsite location.

This sequence ensures that the files on the new copy storage pool volumes are sent offsite, and are not inadvertently kept onsite because of reclamation.

Using Storage on Another Server for Copy Storage Pools: Another resolution to this problem of partially filled volumes is to use storage on another TSM server (device type of SERVER) for storage pool backups. If the other server is at a different site, the copy storage pool volumes are already offsite, with no moving of physical volumes between the sites. See “Using Virtual Volumes to Store Data on Another Server” on page 469 for more information.

Delaying Reuse of Reclaimed Volumes

You should delay the reuse of any reclaimed volumes in copy storage pools for as long as you keep your oldest database backup. Delaying reuse may help you to recover data under certain conditions during recovery from a disaster. For more information on delaying volume reuse, see “Delaying Reuse of Sequential Access Volumes” on page 509.

How Collocation Affects Reclamation

If collocation is enabled and reclamation occurs, the server tries to reclaim the files for each client node or client file space onto a minimal number of volumes. Therefore, if the volumes are manually mounted, the mount operators must:

- Be aware that a tape volume may be rewound more than once if the server completes a separate pass to move the data for each client node or client file space.
- Mount and dismount multiple volumes to allow the server to select the most appropriate volume on which to move data for each client node or client file space. The server tries to select a volume in the following order:
  1. A volume that already contains files belonging to the client file space or client node
  2. An empty volume
  3. The volume with the most available space
  4. Any available volume

If collocation is disabled and reclamation occurs, the server tries to move usable data to new volumes by using the following volume selection criteria, in the order shown:

1. The volume that contains the most data
2. Any partially full volume
3. An empty predefined volume

4. An empty scratch volume

Estimating Space Needs for Storage Pools

This section provides guidelines for estimating the initial storage space required for your installation. You have the following default random access (disk) storage pools available at installation:

- BACKUPPOOL for backed-up files
- ARCHIVEPOOL for archived files
- SPACEMGPOOL for files migrated from client nodes (space-managed files)

You can add space to these storage pools by adding volumes, or you can define additional storage pools.

As your storage environment grows, you may want to consider how policy and storage pool definitions affect where workstation files are stored. Then you can define and maintain multiple storage pools in a hierarchy that allows you to control storage costs by using sequential access storage pools in addition to disk storage pools, and still provide appropriate levels of service to users.

To help you determine how to adjust your policies and storage pools, get information about how much storage is being used (by client node) and for what purposes in your existing storage pools. For more information on how to do this, see “Requesting Information on the Use of Storage Space” on page 209.

Estimating Space Needs in Random Access Storage Pools

To estimate the amount of storage space required for each random access (disk) storage pool:

- Determine the amount of disk space needed for different purposes:
  - For backup storage pools, provide enough disk space to support efficient daily incremental backups.
  - For archive storage pools, provide sufficient space for a user to archive a moderate size file system without causing migration from the disk storage pool to occur.
  - For storage pools for space-managed files, provide enough disk space to support the daily space-management load from HSM clients, without causing migration from the disk storage pool to occur.
- Decide what percentage of this data you want to keep on disk storage space. Establish migration thresholds to have the server automatically migrate the remainder of the data to less expensive storage media in sequential access storage pools.

  See “Choosing Appropriate Migration Threshold Values” on page 178 for recommendations on setting migration thresholds.

Estimating Space for Backed-Up Files in a Random Access Storage Pool

To estimate the total amount of space needed for all backed-up files stored in a single random access (disk) storage pool, use the following formula:

```
Backup space = WkstSize * Utilization * VersionExpansion * NumWkst
```

where:

Backup Space
: The total amount of storage pool disk space needed.

WkstSize
The average data storage capacity of a workstation. For example, if the typical workstation at your installation has a 4GB hard drive, then the average workstation storage capacity is 4GB.

Utilization
An estimate of the fraction of each workstation disk space used, in the range 0 to 1. For example, if you expect that disks on workstations are 75% full, then use 0.75.

VersionExpansion
An expansion factor (greater than 1) that takes into account the additional backup versions, as defined in the copy group. A rough estimate allows 5% additional files for each backup copy. For example, for a version limit of 2, use 1.05, and for a version limit of 3, use 1.10.

NumWkst
The estimated total number of workstations that the server supports.

If clients use compression, the amount of space required may be less than the amount calculated, depending on whether the data is compressible.

Estimating Space for Archived Files in a Random Access Storage Pool

Estimating the amount of storage space for archived files is more difficult, because the number of archived files generated by users is not necessarily related to the amount of data stored on their workstations.

To estimate the total amount of space needed for all archived files in a single random access (disk) storage pool, determine what percentage of user files are typically archived.

Work with policy administrators to calculate this percentage based on the number and type of archive copy groups defined. For example, if policy administrators have defined archive copy groups for only half of the policy domains in your enterprise, then estimate that you need less than 50% of the amount of space you have defined for backed-up files.

Because additional storage space can be added at any time, you can start with a modest amount of storage space and increase the space by adding storage volumes to the archive storage pool, as required.

Estimating Space Needs in Sequential Access Storage Pools

To estimate the amount of space required for sequential access storage pools, consider:

- The amount of data being migrated from disk storage pools
- The length of time backed-up files are retained, as defined in backup copy groups
- The length of time archived files are retained, as defined in archive copy groups
- How frequently you reclaim unused space on sequential volumes

  See “Reclaiming Space in Sequential Access Storage Pools” on page 189 for information about setting a reclamation threshold.
- Whether or not you use collocation to reduce the number of volume mounts required when restoring or retrieving large numbers of files from sequential volumes

If you use collocation, you may need additional tape drives and volumes.

See “Keeping a Client’s Files Together: Collocation” on page 184 for information about using collocation for your storage pools.

- The type of storage devices and sequential volumes supported at your installation

Monitoring Storage Pools and Volumes

Any administrator can query for information about a storage pool by viewing a standard or a detailed report. Use these reports to monitor storage pool usage, including:

- Whether you need to add space to your disk and sequential access storage pools
- The status of the process of migrating data from one to storage pool to the next storage pool in the storage hierarchy
- The use of disk space by cached copies of files that have been migrated to the next storage pool

Monitoring Space Available in a Storage Pool

Monitoring the space available in storage pools is important to ensure that client operations such as backup can complete successfully. To make more space available, you may need to define more volumes for disk storage pools, or add more volumes for sequential access storage pools such as tape. For more information on maintaining a supply of volumes in libraries, see “Managing the Volume Inventory” on page 120.

To request a standard report that shows all storage pools defined to the system, enter:

```
query stgpool
```

Figure 24 shows a standard report with all storage pools defined to the system. To monitor the use of storage pool space, review the *Estimated Capacity* and *Pct Util* columns.

```
Storage      Device       Estimated    Pct    Pct  High   Low  Next
Pool Name    Class Name    Capacity   Util   Migr   Mig   Mig  Storage
                               (MB)                 Pct   Pct  Pool
-----------  ----------  ----------  -----  -----  ----  ----  -----------
ARCHIVEPOOL  DISK               0.0    0.0    0.0    90    70
BACKTAPE     TAPE             180.0   85.0  100.0    90    70
BACKUPPOOL   DISK              80.0   51.6   51.6    50    30  BACKTAPE
COPYPOOL     TAPE             300.0   42.0
ENGBACK1     DISK               0.0    0.0    0.0    85    40  BACKTAPE
```

Figure 24. Information about Storage Pools

Estimated Capacity

Specifies the space available in the storage pool in megabytes.

For a disk storage pool, this value reflects the total amount of available space in the storage pool, including any volumes that are varied offline.

For a sequential access storage pool, this value is an estimate of the total amount of available space on all volumes in the storage pool. The total includes volumes with any access mode (read-write, unavailable, read-only, offsite, or destroyed). The total includes scratch volumes that the storage pool can acquire only when the storage pool is using at least one scratch volume for data.

Volumes in a sequential access storage pool, unlike those in a disk storage pool, do not contain a precisely known amount of space. Data is written to a volume as necessary until the end of the volume is reached. For this reason, the estimated capacity is truly an *estimate* of the amount of available space in a sequential access storage pool.

Pct Util

Specifies, as a percentage, the space used in each storage pool.

For disk storage pools, this value reflects the total number of disk blocks currently allocated by TSM. Space is allocated for backed-up, archived, or space-managed files that are eligible for server migration, cached files that are copies of server-migrated files, and files that reside on any volumes that are varied offline.

Note: The value for Pct Util can be higher than the value for Pct Migr if you query for storage pool information while a client transaction (such as a backup) is in progress. The value for Pct Util is determined by the amount of space actually allocated (while the transaction is in progress). The value for Pct Migr represents only the space occupied by *committed* files. At the end of the transaction, Pct Util and Pct Migr become synchronized.

For sequential access storage pools, this value is the percentage of the total bytes of storage available that are currently being used to store active data (data that is not expired). Because the server can only estimate the available capacity of a sequential access storage pool, this percentage also reflects an estimate of the actual utilization of the storage pool.

Example: Monitoring the Capacity of a Backup Storage Pool

Figure 24 on page 198 shows that the estimated capacity for a disk storage pool named BACKUPPOOL is 80MB, which is the amount of available space on disk storage. More than half (51.6%) of the available space is occupied by either backup files or cached copies of backup files.

The estimated capacity for the tape storage pool named BACKTAPE is 180MB, which is the total estimated space available on all tape volumes in the storage pool. This report shows that 85% of the estimated space is currently being used to store workstation files.

Note: This report also shows that volumes have not yet been defined to the ARCHIVEPOOL and ENGBACK1 storage pools, because the storage pools show an estimated capacity of 0.0MB.

Monitoring the Use of Storage Pool Volumes

Task	Required Privilege Class
Display information about volumes	Any administrator

You can query the server for information about storage pool volumes:

- General information about a volume, such as the following:
  - Current access mode and status of the volume

  - Amount of available space on the volume
  - Location
- Contents of a storage pool volume (user files on the volume)
- The volumes that are used by a client node

Getting General Information about Storage Pool Volumes

To request general information about all volumes defined to the server, enter:

```
query volume
```

Figure 25 shows an example of the output of this standard query. The example illustrates that data is being stored on the 8mm tape volume named WREN01, as well as on several other volumes in various storage pools.

To query the server for a detailed report on volume WREN01 in the storage pool named TAPEPOOL, enter:

```
query volume wren01 format=detailed
```

```
Volume Name               Storage      Device      Estimated    Pct   Volume
                          Pool Name    Class Name   Capacity   Util   Status
                                                        (MB)
------------------------  -----------  ----------  ---------  -----  --------
/dev/raixvol1             AIXPOOL1     DISK            240.0   26.3  On-Line
/dev/raixvol2             AIXPOOL2     DISK            240.0   36.9  On-Line
/dev/rdosvol1             DOSPOOL1     DISK            240.0   72.2  On-Line
/dev/rdosvol2             DOSPOOL2     DISK            240.0   74.1  On-Line
/dev/ros2vol1             OS2POOL1     DISK            240.0   55.7  On-Line
/dev/ros2vol2             OS2POOL2     DISK            240.0   51.0  On-Line
WREN00                    TAPEPOOL     TAPE8MM       2,472.0    0.0  Filling
WREN01                    TAPEPOOL     TAPE8MM       2,472.0    2.2  Filling
```

Figure 25. Information about Storage Pool Volumes

Figure 26 shows the output of this detailed query. Table 18 on page 201 gives some suggestions on how you can use the information.

```
                  Volume Name: WREN01
            Storage Pool Name: TAPEPOOL
            Device Class Name: TAPE8MM
      Estimated Capacity (MB): 2,472.0
                     Pct Util: 26.3
                Volume Status: Filling
                       Access: Read/Write
       Pct. Reclaimable Space: 5.3
              Scratch Volume?: No
              In Error State?: No
     Number of Writable Sides: 1
      Number of Times Mounted: 4
            Write Pass Number: 2
    Approx. Date Last Written: 12/04/1996 11:33:26
       Approx. Date Last Read: 12/03/1996 16:42:55
          Date Became Pending:
       Number of Write Errors: 0
        Number of Read Errors: 0
              Volume Location:
Last Update by (administrator): TANAGER
        Last Update Date/Time: 12/02/1996 13:20:14
```

Figure 26. Detailed Information for a Storage Pool Volume

Table 18. Using the Detailed Report for a Volume

Task	Fields and Description
Ensure the volume is available.	*Volume Status* *Access*
	Check the *Volume Status* to see if a disk volume has been varied offline, or if a sequential access volume is currently being filled with data. Check the *Access* to determine whether files can be read from or written to this volume.
Monitor the use of storage space.	*Estimated Capacity* *Pct Util*
	The *Estimated Capacity* is determined by the device class associated with the storage pool to which this volume belongs. Based on the estimated capacity, the system tracks the percentage of space occupied by client files (*Pct Util*). In this example, 26.3% of the estimated capacity is currently in use.
Monitor the error status of the volume.	*Number of Write Errors* *Number of Read Errors*
	The server reports when the volume is in an error state and automatically updates the access mode of the volume to read-only. The *Number of Write Errors* and *Number of Read Errors* indicate the type and severity of the problem. Audit a volume when it is placed in error state. See "Auditing a Storage Pool Volume" on page 529 for information about auditing a volume.
Monitor the life of sequential access volumes that you have defined to the storage pool.	*Scratch Volume?* *Write Pass Number* *Number of Times Mounted* *Approx. Date Last Written* *Approx. Date Last Read*
	The server maintains usage statistics on volumes that are defined to storage pools. Statistics on a volume explicitly defined by an administrator remain for as long as the volume is defined to the storage pool. The server continues to maintain the statistics on defined volumes even as the volume is reclaimed and reused. However, the server deletes the statistics on the usage of a scratch volume when the volume returns to scratch status (after reclamation or after all files are deleted from the volume). In this example, WREN01 is a volume defined to the server by an administrator, not a scratch volume (*Scratch Volume?* is *No*). The *Write Pass Number* indicates the number of times the volume has been written to, starting from the beginning of the volume. A value of one indicates that a volume is being used for the first time. In this example, WREN01 has a write pass number of two, which indicates space on this volume may have been reclaimed or deleted once before. Compare this value to the specifications provided with the media that you are using. The manufacturer may recommend a maximum number of write passes for some types of tape media. You may need to retire your tape volumes after reaching the maximum passes to better ensure the integrity of your data. To retire a volume, move the data off the volume by using the MOVE DATA command. See "Moving Files from One Volume to Another Volume" on page 211. Use the *Number of Times Mounted*, the *Approx. Date Last Written*, and the *Approx. Date Last Read* to help you estimate the life of the volume. For example, if more than six months have passed since the last time this volume has been written to or read from, audit the volume to ensure that files can still be accessed. See "Auditing a Storage Pool Volume" on page 529 for information about auditing a volume.

Table 18. Using the Detailed Report for a Volume (continued)

Task	Fields and Description
Determine the location of a volume in a sequential access storage pool.	*Location* When you define or update a sequential access volume, you can give location information for the volume. The detailed query displays this location name. The location information can be useful to help you track volumes, for example, offsite volumes in copy storage pools.
Determine if a volume in a sequential access storage pool is waiting for the reuse delay period to expire.	*Date Became Pending* A sequential access volume is placed in the pending state after the last file is deleted or moved from the volume. All the files that the pending volume had contained were expired or deleted, or were moved from the volume. Volumes remain in the pending state for as long as specified with the REUSEDELAY parameter for the storage pool to which the volume belongs.

Whether or not a volume is full, at times the Pct Util (percent of the volume utilized) plus the Pct Reclaimable Space (percent of the volume that can be reclaimed) may add up to more than 100 percent. This can happen when a volume contains aggregates that have empty space because of files in the aggregates that have expired or been deleted. The Pct Util field shows all space occupied by both non-aggregated files and aggregates, including empty space within aggregates. The Pct Reclaimable Space field includes any space that is reclaimable on the volume, also including empty space within aggregates. Because both fields include the empty space within aggregates, these values may add up to more than 100 percent. For more information about aggregates, see “How the Server Groups Files before Storing” on page 171 and “Requesting Information on the Use of Storage Space” on page 209.

Getting Information about the Contents of a Storage Pool Volume

Any administrator can request information about the contents of a storage pool volume. Viewing the contents of a storage volume is useful when a volume is damaged or before you do the following:

- Request the server to correct any inconsistencies (AUDIT VOLUME command)
- Move files from one volume to other volumes
- Delete a volume from a storage pool

Because the server tracks the contents of a storage volume through its database, the server does not need to access the requested volume to determine its contents.

The report generated by a QUERY CONTENT command shows the contents of a volume. This report can be extremely large and may take a long time to produce. To reduce the size of this report, narrow your search by selecting one or all of the following search criteria:

Node name
Name of the node whose files you want to include in the query.

File space name
Names of file spaces to include in the query. File space names are case-sensitive and must be entered exactly as they are known to the server. Use the QUERY FILESPACE command to find the correct capitalization.

Number of files to be displayed
Enter a positive integer, such as 10, to list the first ten files stored on the volume. Enter a negative integer, such as -15, to list the last fifteen files stored on the volume.

Filetype
Specifies which types of files, that is, backup versions, archive copies, or space-managed files, or a combination of these.

Format of how the information is displayed
Standard or detailed information for the specified volume.

Damaged
Specifies whether to restrict the query output either to files that are known to be damaged, or to files that are not known to be damaged.

Copied
Specifies whether to restrict the query output to either files that are backed up to a copy storage pool, or to files that are not backed up to a copy storage pool.

Viewing a Standard Report on the Contents of a Volume: To view the first seven backup files on volume WREN01 from file space /usr on client node TOMC, for example, enter:

```
query content wren01 node=tomc filespace=/usr count=7 type=backup
```

Figure 27 displays a standard report which shows the first seven files from file space /usr on TOMC stored in WREN01.

```
Node Name                Type Filespace  Client's Name for File
                              Name
------------------------ ---- ---------- --------------------------------------
TOMC                     Bkup /usr       /bin/ acctcom
TOMC                     Bkup /usr       /bin/ acledit
TOMC                     Bkup /usr       /bin/ aclput
TOMC                     Bkup /usr       /bin/ admin
TOMC                     Bkup /usr       /bin/ ar
TOMC                     Bkup /usr       /bin/ arcv
TOMC                     Bkup /usr       /bin/ banner
```

Figure 27. A Standard Report on the Contents of a Volume

The report lists logical files on the volume. If a file on the volume is an aggregate of logical files (backed-up or archived client files), all logical files that are part of the aggregate are included in the report. An aggregate file can be stored on more than one volume, and therefore not all of the logical files in the report may actually be stored on the volume being queried.

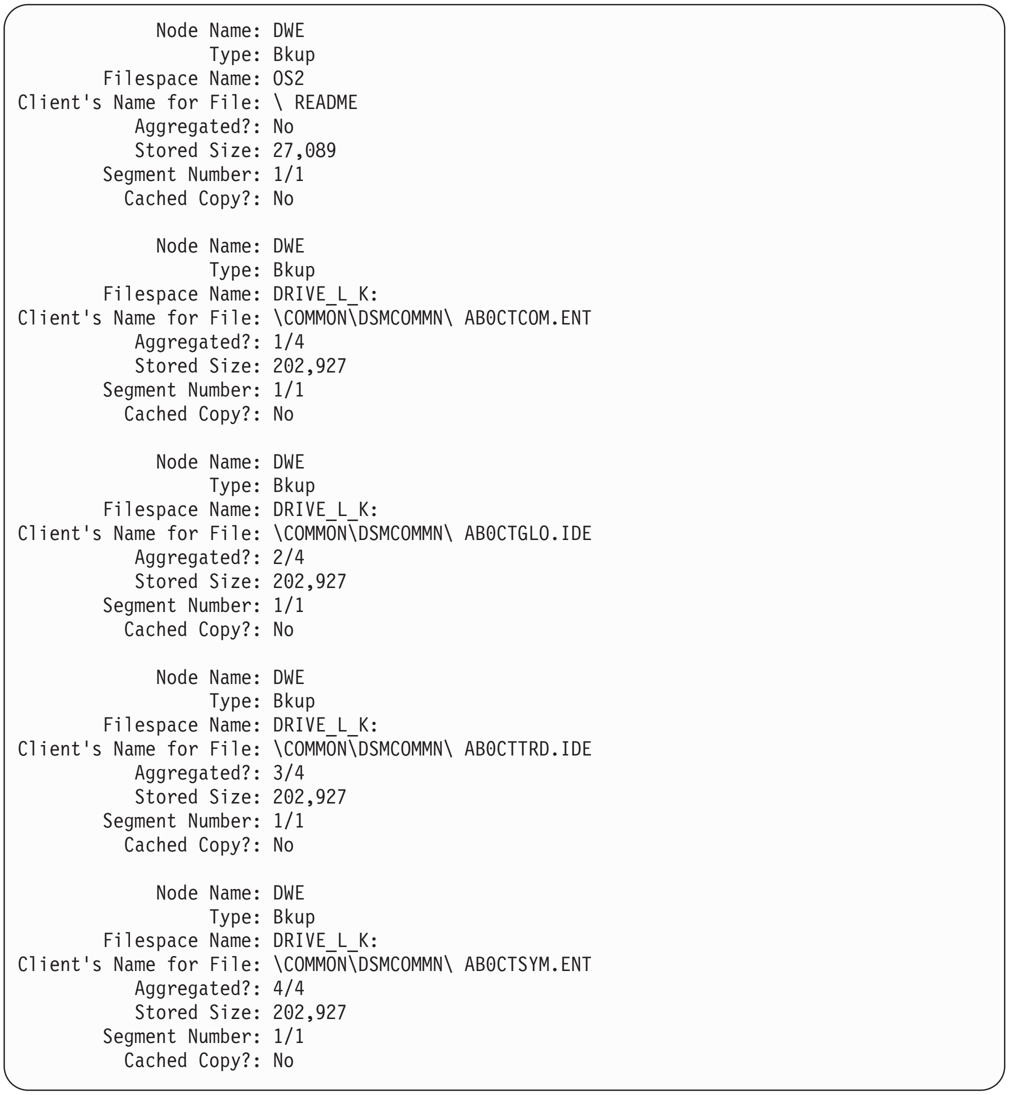
Viewing a Detailed Report on the Contents of a Volume: To display detailed information about the files stored on volume VOL1, enter:

```
query content vol1 format=detailed
```

Figure 28 on page 204 displays a detailed report that shows the files stored on VOL1. The report lists logical files and shows whether each file is part of an aggregate file. If a logical file is stored as part of an aggregate file, the information in the **Segment Number**, **Stored Size**, and **Cached Copy?** fields apply to the aggregate, not to the individual logical file.

If a logical file is part of an aggregate file, the **Aggregated?** field shows the sequence number of the logical file within the aggregate file. For example, the **Aggregated?** field contains the value 2/4 for the file AB0CTGLO.IDE, meaning that this file is the second of four files in the aggregate. All logical files that are part of an aggregate are included in the report. An aggregate file can be stored on more than one volume, and therefore not all of the logical files in the report may actually be stored on the volume being queried.

For disk volumes, the **Cached Copy?** field identifies whether the file is a cached copy of a file that has been migrated to the next storage pool in the hierarchy.



```
             Node Name: DWE
                  Type: Bkup
        Filespace Name: OS2
Client's Name for File: \ README
           Aggregated?: No
           Stored Size: 27,089
        Segment Number: 1/1
          Cached Copy?: No

             Node Name: DWE
                  Type: Bkup
        Filespace Name: DRIVE_L_K:
Client's Name for File: \COMMON\DSMCOMMN\ AB0CTCOM.ENT
           Aggregated?: 1/4
           Stored Size: 202,927
        Segment Number: 1/1
          Cached Copy?: No

             Node Name: DWE
                  Type: Bkup
        Filespace Name: DRIVE_L_K:
Client's Name for File: \COMMON\DSMCOMMN\ AB0CTGLO.IDE
           Aggregated?: 2/4
           Stored Size: 202,927
        Segment Number: 1/1
          Cached Copy?: No

             Node Name: DWE
                  Type: Bkup
        Filespace Name: DRIVE_L_K:
Client's Name for File: \COMMON\DSMCOMMN\ AB0CTTRD.IDE
           Aggregated?: 3/4
           Stored Size: 202,927
        Segment Number: 1/1
          Cached Copy?: No

             Node Name: DWE
                  Type: Bkup
        Filespace Name: DRIVE_L_K:
Client's Name for File: \COMMON\DSMCOMMN\ AB0CTSYM.ENT
           Aggregated?: 4/4
           Stored Size: 202,927
        Segment Number: 1/1
          Cached Copy?: No
```

Figure 28. Viewing a Detailed Report of the Contents of a Volume

Finding the Volumes Used by a Client Node

You can use the server’s SELECT command to find the sequential volumes used by a client node. Use SELECT to perform an SQL query of the VOLUMEUSAGE table in the TSM database. For example, to get a list of volumes used by the EXCH1 client node in the TAPEPOOL storage pool, enter the following command:

```
select volume_name from volumeusage where node_name='EXCH1' and
stgpool_name='TAPEPOOL'
```

The results are something like the following:

```
VOLUME_NAME
------------------
TAPE01
TAPE08
TAPE13
TAPE21
```

For more information about using the SELECT command, see *Administrator's Reference*.

Monitoring Migration Processes

Four fields on the standard storage pool report provide you with information about the migration process. They include:

Pct Migr
Specifies the percentage of data in each storage pool that can be migrated. This value is used to determine when to start or stop migration.

For disk storage pools, this value represents the amount of disk space occupied by backed-up, archived, or space-managed files that can be migrated to another storage pool, including files on volumes that are varied offline. Cached data are excluded in the Pct Migr value.

For sequential access storage pools, this value is the percentage of the total volumes in the storage pool that actually contain data at the moment. For example, assume a storage pool has four explicitly defined volumes, and a maximum scratch value of six volumes. If only two volumes actually contain data at the moment, then Pct Migr will be 20%.

This field is blank for copy storage pools.

High Mig Pct
Specifies when the server can begin migrating data from this storage pool. Migration can begin when the percentage of data that can be migrated reaches this threshold. (This field is blank for copy storage pools.)

Low Mig Pct
Specifies when the server can stop migrating data from this storage pool. Migration can end when the percentage of data that can be migrated falls below this threshold. (This field is blank for copy storage pools.)

Next Storage Pool
Specifies the primary storage pool destination to which data is migrated. (This field is blank for copy storage pools.)

Example: Monitoring the Migration of Data Between Storage Pools

Figure 24 on page 198 shows that the migration thresholds for BACKUPPOOL storage pool are set to 50% for the *high migration threshold* and 30% for the *low migration threshold*.

When the amount of migratable data stored in the BACKUPPOOL storage pool reaches 50%, the server can begin to migrate files to BACKTAPE.

To monitor the migration of files from BACKUPPOOL to BACKTAPE, enter:

```
query stgpool back*
```

See Figure 29 on page 206 for an example of the results of this command.

If caching is on for a disk storage pool and files are migrated, the Pct Util value does not change because the cached files still occupy space in the disk storage pool. However, the Pct Migr value decreases because the space occupied by cached files is no longer migratable.

```
Storage      Device       Estimated    Pct    Pct  High   Low  Next
Pool Name    Class Name    Capacity   Util   Migr   Mig   Mig  Storage
                               (MB)                 Pct   Pct  Pool
-----------  ----------  ----------  -----  -----  ----  ----  -----------
BACKTAPE     TAPE             180.0   95.2  100.0    90    70
BACKUPPOOL   DISK              80.0   51.6   28.8    50    30  BACKTAPE
```

Figure 29. Information on Backup Storage Pools

You can query the server to monitor the migration process by entering:

```
query process
```

A message similar to Figure 30 is displayed:

```
 Process Process Description      Status
  Number
-------- ------------------------ ---------------------------------------------
       2 Migration                Disk Storage Pool BACKUPPOOL, Moved Files:
                                   1086, Moved Bytes: 25555579, Unreadable
                                   Files: 0, Unreadable Bytes: 0
```

Figure 30. Information on the Migration Process

When migration is finished, the server displays the following message:

```
ANR1101I Migration ended for storage pool BACKUPPOOL.
```

Handling Problems during the Migration Process

A problem can occur that causes the migration process to be suspended. For example, there may not be sufficient space in the storage pool to which data is being migrated. When migration is suspended, the process might be retried.

At this point, a system administrator can:

- Cancel the migration process. See "Canceling the Migration Process" for additional information.
- End the migration process by changing the attributes of the storage pool from which data is being migrated. See "Ending the Migration Process by Changing Storage Pool Characteristics" on page 207 for additional information.
- Provide additional space. See "Providing Additional Space for the Migration Process" on page 207 for additional information.

The server attempts to restart the migration process every 60 seconds for several minutes and if not successful will terminate the migration process.

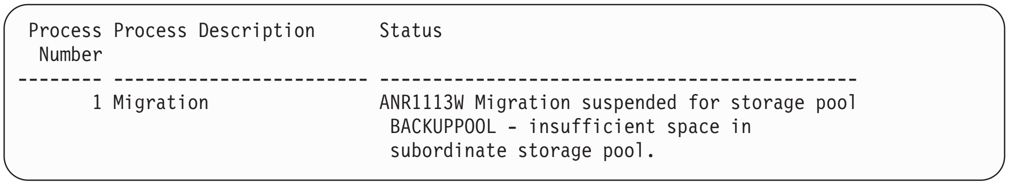
Canceling the Migration Process

To stop server migration when a problem occurs or when you need the resources the process is using, you can cancel the migration.

First determine the identification number of the migration process by entering:

```
query process
```

A message similar to Figure 31 is displayed:



```
 Process Process Description      Status
  Number
-------- ------------------------ ---------------------------------------------
       1 Migration                ANR1113W Migration suspended for storage pool
                                   BACKUPPOOL - insufficient space in
                                   subordinate storage pool.
```

Figure 31. Getting the Identification Number of the Migration Process

Then you can cancel the migration process by entering:

```
cancel process 1
```

Ending the Migration Process by Changing Storage Pool Characteristics

Some errors cause the server to continue attempting to restart the migration process after 60 seconds. (If the problem still exists after several minutes, the migration process will end.) To stop the repeated attempts at restart, you can change some characteristics of the storage pool from which data is being migrated. Depending on your environment, you can:

- Set higher migration thresholds for the storage pool from which data is being migrated. The higher threshold means the storage pool must have more migratable data before migration starts. This change delays migration.

  In the example in "Example: Monitoring the Migration of Data Between Storage Pools" on page 205, you could update the disk storage pool BACKUPPOOL.
- Add volumes to the pool from which data is being migrated. Adding volumes decreases the percentage of data that is migratable (Pct Migr).

  In the example in "Example: Monitoring the Migration of Data Between Storage Pools" on page 205, you could add volumes to the disk storage pool BACKUPPOOL to increase its storage capacity.

  Note: Do this only if you received an out-of-space message for the storage pool to which data is being migrated.

Providing Additional Space for the Migration Process

A migration process can be suspended because of insufficient space in the storage pool to which data is being migrated. To allow the migration process to complete, you can provide additional storage volumes for that storage pool.

In the example in "Example: Monitoring the Migration of Data Between Storage Pools" on page 205, you could add volumes to the BACKTAPE storage pool or increase the maximum number of scratch tapes allowed for it. Either way, you increase the storage capacity of BACKTAPE.

Monitoring the Use of Cache Space on Disk Storage

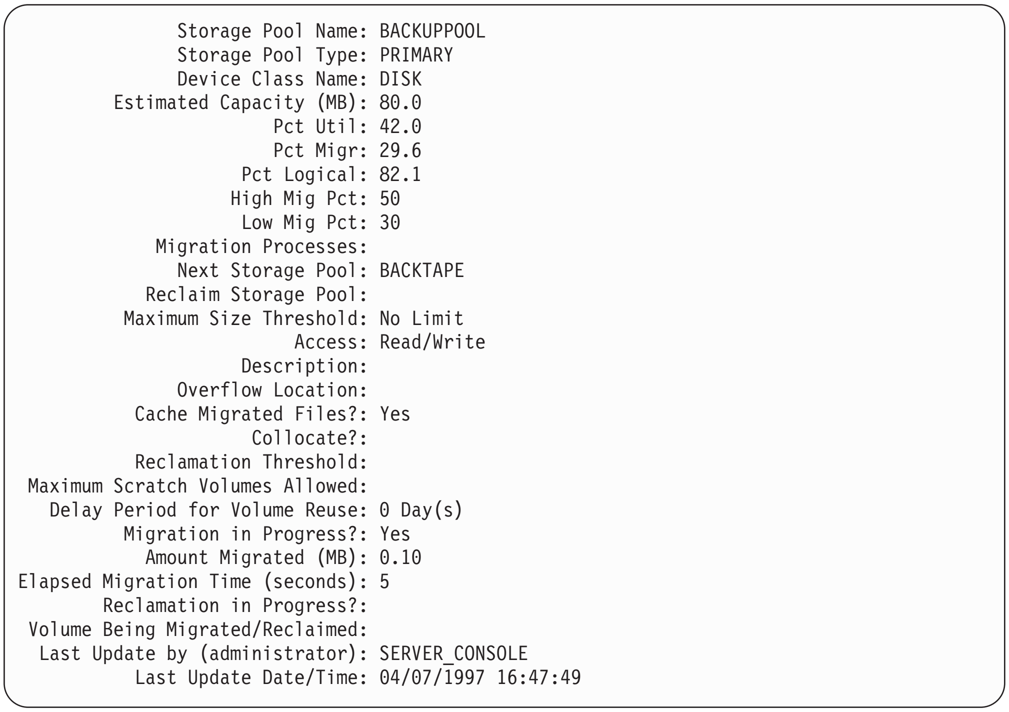
The Pct Util value includes cached data on a volume (when cache is enabled) and the Pct Migr value excludes cached data. Therefore, when cache is enabled and migration occurs, the Pct Migr value decreases while the Pct Util value remains the same. The Pct Util value remains the same because the migrated data remains on the volume as cached data. In this case, the Pct Util value only decreases when the cached data expires.

If you update a storage pool from CACHE=YES to CACHE=NO, the cached files will not disappear immediately. The Pct Util value will be unchanged. The cache space will be reclaimed over time as the server needs the space, and no additional cached files will be created.

To determine whether cache is being used on disk storage and to monitor how much space is being used by cached copies, query the server for a detailed storage pool report. For example, to request a detailed report for BACKUPPOOL, enter:

```
query stgpool backuppool format=detailed
```

Figure 32 displays a detailed report for the storage pool.



```
               Storage Pool Name: BACKUPPOOL
               Storage Pool Type: PRIMARY
               Device Class Name: DISK
         Estimated Capacity (MB): 80.0
                        Pct Util: 42.0
                        Pct Migr: 29.6
                     Pct Logical: 82.1
                    High Mig Pct: 50
                     Low Mig Pct: 30
             Migration Processes:
               Next Storage Pool: BACKTAPE
            Reclaim Storage Pool:
          Maximum Size Threshold: No Limit
                          Access: Read/Write
                     Description:
               Overflow Location:
           Cache Migrated Files?: Yes
                      Collocate?:
           Reclamation Threshold:
 Maximum Scratch Volumes Allowed:
   Delay Period for Volume Reuse: 0 Day(s)
          Migration in Progress?: Yes
            Amount Migrated (MB): 0.10
Elapsed Migration Time (seconds): 5
        Reclamation in Progress?:
 Volume Being Migrated/Reclaimed:
  Last Update by (administrator): SERVER_CONSOLE
           Last Update Date/Time: 04/07/1997 16:47:49
```

Figure 32. Detailed Storage Pool Report

When **Cache Migrated Files?** is set to **Yes**, the value for Pct Util should not change because of migration, because cached copies of files migrated to the next storage pool remain in disk storage.

This example shows that utilization remains at 42%, even after files have been migrated to the BACKTAPE storage pool, and the current amount of data eligible for migration is 29.6%.

When **Cache Migrated Files?** is set to **No**, the value for Pct Util more closely matches the value for Pct Migr because cached copies are not retained in disk storage.

Requesting Information on the Use of Storage Space

Task	Required Privilege Class
Query the server for information about server storage	Any administrator

Any administrator can request information about server storage occupancy. Use the QUERY OCCUPANCY command for reports with information broken out by node or file space. Use this report to determine the amount of space used by:

- Client node and file space
- Storage pool or device class
- Type of data (backup, archive, or space-managed)

Each report gives two measures of the space in use by a storage pool:

- Logical space occupied

  The amount of space used for logical files. A logical file is a client file. A logical file is stored either as a single physical file, or in an aggregate with other logical files.
- Physical space occupied

  The amount of space used for physical files. A physical file is either a single logical file, or an aggregate file composed of logical files.

  An aggregate file may contain empty space that had been used by logical files that are now expired or deleted. Therefore, the amount of space used by physical files is equal to or greater than the space used by logical files. The difference gives you a measure of how much unused space any aggregate files may have. The unused space can be reclaimed in sequential storage pools.

You can also use this report to evaluate the average size of workstation files stored in server storage.

Amount of Space Used by Client Node

Any administrator can request information about the space used by each client node and file space:

- How much data has been backed up, archived, or migrated to server storage
- How many of the files that are in server storage have been backed up to a copy storage pool
- The amount of storage space being used

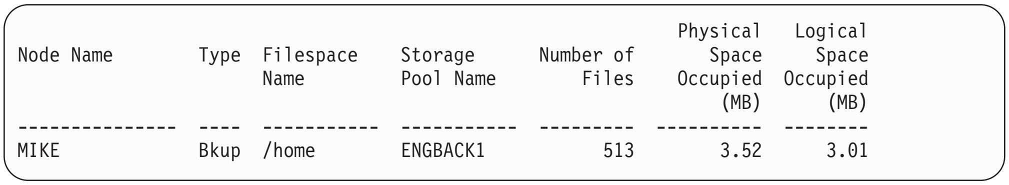
To determine the amount of server storage space used by the /home file space belonging to the client node MIKE, for example, enter:

```
query occupancy mike /home
```

Remember that file space names are case-sensitive and must be entered exactly as they are known to the server. Use the QUERY FILESPACE command to determine the correct capitalization. For more information, see “Managing File Spaces” on page 242.

Figure 33 on page 210 shows the results of the query. The report shows the number of files backed up, archived, or migrated from the /home file space belonging to MIKE. The report also shows how much space is occupied in each storage pool.

If you back up the ENGBACK1 storage pool to a copy storage pool, the copy storage pool would also be listed in the report. To determine how many of the client node’s files in the primary storage pool have been backed up to a copy storage pool, compare the number of files in each pool type for the client node.



```
                                                          Physical   Logical
Node Name        Type  Filespace    Storage      Number of     Space     Space
                       Name         Pool Name        Files  Occupied  Occupied
                                                                (MB)      (MB)
---------------  ----  -----------  -----------  ---------  ----------  --------
MIKE             Bkup  /home        ENGBACK1           513        3.52      3.01
```

Figure 33. A Report of the Occupancy of Storage Pools by Client Node

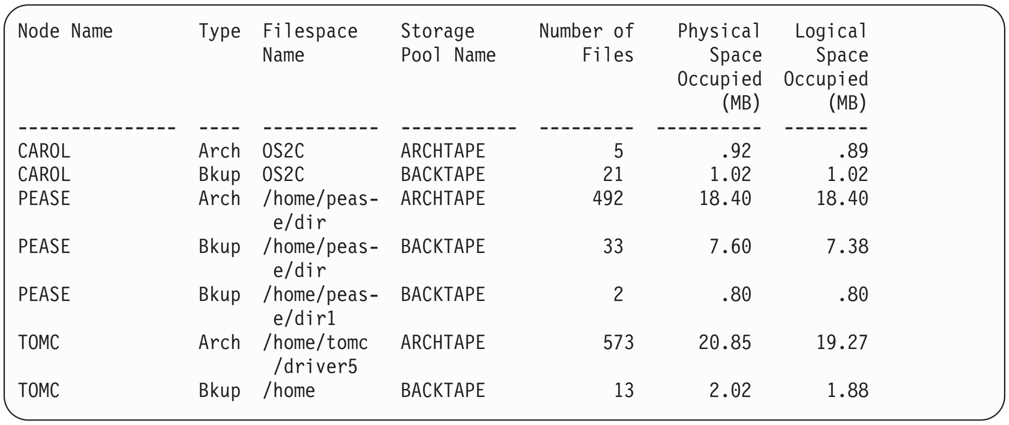
Amount of Space Used by Storage Pool or Device Class

You can monitor the amount of space being used by an individual storage pool, a group of storage pools, or storage pools categorized by a particular device class. Creating occupancy reports on a regular basis can help you with capacity planning.

To query the server for the amount of data stored in backup tape storage pools belonging to the TAPECLASS device class, for example, enter:

```
query occupancy devclass=tapeclass
```

Figure 34 displays a report on the occupancy of tape storage pools assigned to the TAPECLASS device class.



```
Node Name        Type  Filespace    Storage      Number of    Physical   Logical
                       Name         Pool Name        Files       Space     Space
                                                              Occupied  Occupied
                                                                  (MB)      (MB)
---------------  ----  -----------  -----------  ---------  ----------  --------
CAROL            Arch  OS2C         ARCHTAPE             5         .92       .89
CAROL            Bkup  OS2C         BACKTAPE            21        1.02      1.02
PEASE            Arch  /home/peas-  ARCHTAPE           492       18.40     18.40
                        e/dir
PEASE            Bkup  /home/peas-  BACKTAPE            33        7.60      7.38
                        e/dir
PEASE            Bkup  /home/peas-  BACKTAPE             2         .80       .80
                        e/dir1
TOMC             Arch  /home/tomc   ARCHTAPE           573       20.85     19.27
                        /driver5
TOMC             Bkup  /home        BACKTAPE            13        2.02      1.88
```

Figure 34. A Report on the Occupancy of Storage Pools by Device Class

Note: For archived data, you may see “(archive)” in the Filespace Name column instead of a file space name. This means that the data was archived before collocation by file space was supported by the server.

Amount of Space Used by Backed-Up, Archived, or Space-Managed Files

You can query the server for the amount of space used by backed-up, archived, and space-managed files. By determining the average size of workstation files stored in server storage, you can estimate how much storage capacity you might need when registering new client nodes to the server. See “Estimating Space Needs for Storage Pools" on page 196 and "Estimating Space for Archived Files in a Random Access Storage Pool" on page 197 for information about planning storage space.

To request a report about backup versions stored in the disk storage pool named BACKUPPOOL, for example, enter:

```
query occupancy stgpool=backuppool type=backup
```

Figure 35 displays a report on the amount of server storage used for backed-up files.

```
Node Name        Type  Filespace    Storage      Number of    Physical   Logical
                       Name         Pool Name        Files       Space     Space
                                                              Occupied  Occupied
                                                                  (MB)      (MB)
---------------  ----  -----------  -----------  ---------  ----------  --------
CAROL            Bkup  OS2C         BACKUPPOOL         513       23.52     23.52
CAROL            Bkup  OS2D         BACKUPPOOL         573       20.85     20.85
PEASE            Bkup  /marketing   BACKUPPOOL         132       12.90      9.01
PEASE            Bkup  /business    BACKUPPOOL         365       13.68      6.18
TOMC             Bkup  /            BACKUPPOOL         177       21.27     21.27
```

Figure 35. A Report of the Occupancy of Backed-Up Files in Storage Pools

To determine the average size of backup versions stored in BACKUPPOOL, complete the following steps using the data provided in Figure 35:

1. Add the number of megabytes of space occupied by backup versions.

   In this example, backup versions occupy 92.22MB of space in BACKUPPOOL.
2. Add the number of files stored in the storage pool.

   In this example, 1760 backup versions reside in BACKUPPOOL.
3. Divide the space occupied by the number of files to determine the average size of each file backed up to the BACKUPPOOL.

   In this example, the average size of each workstation file backed up to BACKUPPOOL is about 0.05MB, or approximately 50KB.

You can use this average to estimate the capacity required for additional storage pools that are defined to the server.

Moving Files from One Volume to Another Volume

You can move files from one volume to another volume in the same or a different storage pool using the MOVE DATA command. The volumes can be onsite volumes or offsite volumes. During normal operations, you do not need to move data. You might need to move data in some situations, for example, when you need to salvage any readable data from a damaged TSM volume.

During the data movement process, the server:

- Moves any readable files to available volumes in the specified destination storage pool
- Deletes any cached copies from a disk volume
- Attempts to bypass any files that previously were marked as damaged

During the data movement process, users cannot access the volume to restore or retrieve files, and no new files can be written to the volume.

Note:

- Files in a copy storage pool do not move when primary files are moved.
- You can only move data for volumes belonging to a storage pool with DATAFORMAT=NATIVE or DATAFORMAT=NONBLOCK.

Task	Required Privilege Class
Move files from a volume in any storage pool to an available volume in any storage pool	System or unrestricted storage
Move files from one volume to an available volume in any storage pool to which you are authorized	Restricted storage

Moving Data to Other Volumes in the Same Storage Pool

Moving files from one volume to other volumes in the same storage pool is useful:

- When you want to free up all space on a volume so that it can be deleted from the TSM server

  See "Deleting Storage Pool Volumes" on page 222 for information about deleting backed-up, archived, or space-managed data before you delete a volume from a storage pool.
- When you need to salvage readable files from a volume that has been damaged
- When you want to delete cached files from disk volumes

  Note: You should not run MOVE DATA and migration processing concurrently on the same storage pool.

  If you want to force the removal of cached files, you can delete them by moving data from one volume to another volume. During the move process, the server deletes cached files remaining on disk volumes.

If you move data between volumes within the same storage pool and you run out of space in the storage pool before all data is moved from the target volume, then you cannot move all the data from the target volume. In this case, consider moving data to available space in another storage pool as described in "Moving Data to Another Storage Pool".

Moving Data to Another Storage Pool

You can move all data from a volume in one storage pool to volumes in another storage pool. When you specify a target storage pool that is different than the source storage pool, the server uses the storage hierarchy to move data if more space is required.

Note: Data cannot be moved from a primary storage pool to a copy storage pool. Data in a copy storage pool cannot be moved to any other storage pool.

You can move data from random access storage pools to sequential access storage pools. For example, if you have a damaged disk volume and you have a limited amount of disk storage space, you could move all files from the disk volume to a tape storage pool. Moving files from a disk volume to a sequential storage pool may require many volume mount operations if the target storage pool is collocated. Ensure that you have sufficient personnel and media to move files from disk to sequential storage.

Moving Data from an Offsite Volume in a Copy Storage Pool

You can move data from offsite volumes without bringing the volumes onsite. Processing of the MOVE DATA command for primary storage pool volumes does not affect copy storage pool files.

Processing of the MOVE DATA command for volumes in copy storage pools is similar to that of primary storage pools, with the following exceptions:

- Most volumes in copy storage pools may be set to an access mode of *offsite*, making them ineligible to be mounted. During processing of the MOVE DATA command, valid files on offsite volumes are copied from the original files in the primary storage pools. In this way, valid files on offsite volumes are copied without having to mount these volumes. These new copies of the files are written to another volume in the copy storage pool.
- With the MOVE DATA command, you can move data from any primary storage pool volume to any primary storage pool. However, you can move data from a copy storage pool volume *only* to another volume within the same copy storage pool.

When you move files from a volume marked as offsite, the server does the following:

1. Determines which files are still active on the volume from which you are moving data
2. Obtains these files from a primary storage pool or from another copy storage pool
3. Copies the files to one or more volumes in the destination copy storage pool

Procedure for Moving Data

1. Before you move files from a volume, complete the following steps:
   - If you want to ensure that no new files are written to a volume after you move data from it, change the volume's access mode to read-only. This prevents the server from filling the volume with data again as soon as data is moved. You might want to do this if you want to delete the volume.

     See "Updating Storage Pool Volumes" on page 167 for information about updating the access mode of a storage pool volume.
   - Ensure sufficient space is available on volumes within the specified destination storage pool by:
     a. Querying the source storage volume to determine how much space is required on other volumes. See "Monitoring the Use of Storage Pool Volumes" on page 199 for information about requesting information about a storage volume.
     b. Querying the specified destination storage pool to ensure there is sufficient capacity to store the files being moved. See "Monitoring Space Available in a Storage Pool" on page 198 for information about querying a storage pool.

     If you need more storage space, define volumes or increase the maximum number of scratch volumes in the specified destination storage pool. See "Defining Storage Pool Volumes" on page 167 for preparing volumes to be used for server storage.
   - If you are moving files from a volume in a sequential storage pool to another volume in the same storage pool, ensure that the mount limit of the device class associated with the storage pool is greater than one.

See “Requesting Information about a Device Class” on page 151 for requesting information about the mount limit value for the device class.

- If you are moving files from a tape volume to a tape storage pool, ensure that the two tape drives required are available.

2. Move the data using the MOVE DATA command.

   For example, to move the files stored in the /dev/vol3 volume to any available volume in the STGTMP1 storage pool, enter:

   ```
   move data /dev/vol3 stgpool=stgtmp1
   ```

   When you move data from a volume, the server starts a background process and sends informational messages, such as:

   ```
   ANR1140I Move Data process started for volume /dev/vol3
   (process ID 32).
   ```

   The command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

   Note:

   - A volume may not be totally empty after a move data operation completes. For example, the server may be unable to relocate one or more files to another volume because of input/output errors on the device or because errors were found in the file. You can delete the volume with DISCARDDATA=YES to delete the volume and any remaining files. The server then deletes the remaining files that had I/O or other errors.
   - You can only move data for volumes belonging to a storage pool with DATAFORMAT=NATIVE or DATAFORMAT=NONBLOCK.

Requesting Information about the Data Movement Process

To request information on the data movement process, enter:

```
query process
```

Figure 36 shows an example of the report that you receive about the data movement process.

```
 Process Process Description  Status
  Number
-------- -------------------- -------------------------------------------------
      32 Move Data            Volume /dev/vol3, (storage pool BACKUPPOOL),
                               Target Pool STGTMP1, Moved Files: 49, Moved
                               Bytes: 9,121,792, Unreadable Files: 0,
                               Unreadable Bytes: 0. Current File (bytes):
                               3,522,560

                               Current output volume: VOL1.
```

Figure 36. Information on the Data Movement Process

Reclaiming Space in Aggregates During Data Movement

Empty space accumulates in a file aggregate as logical files in that aggregate are deleted. During reclamation processing, the aggregate is reconstructed and this empty space is removed. However, you cannot start reclamation processing only for specific volumes. To reconstruct an aggregate for a specific volume, you can issue the MOVE DATA command with the RECONSTRUCT parameter. In this way, you can move data within a sequential-access storage pool without moving any expired files in the aggregates. You may want to do this if the expired files contain sensitive data and must be purged for legal reasons.

For example, to move the files stored in volume /dev/vol3 to any available volume in the STGTMP1 storage pool and reconstruct the aggregates in that volume, enter:

```
move data /dev/vol3 stgpool=stgtmp1 reconstruct=yes
```

Monitoring the Movement of Data between Volumes

You can query the server for volume information to monitor the movement of data between volumes. For example, to see how much data has moved from the source volume in the move operation example, enter:

```
query volume /dev/vol3 stgpool=backuppool
```

Near the beginning of the move process, querying the volume from which data is being moved gives the following results:

```
Volume Name         Storage       Device         Estimated      Pct     Volume
                    Pool Name     Class Name      Capacity     Util     Status
                                                      (MB)
---------------     -----------   ----------     ---------    -----    --------
/dev/vol3           BACKUPPOOL    DISK                15.0     59.9    On-Line
```

Querying the volume to which data is being moved (VOL1, according to the process query output) gives the following results:

```
Volume Name         Storage       Device         Estimated      Pct     Volume
                    Pool Name     Class Name      Capacity     Util     Status
                                                      (MB)
----------------    -----------   ----------     ---------    -----    --------
VOL1                STGTMP1       8500DEV          4,944.0      0.3    Filling
```

At the end of the move process, querying the volume from which data was moved gives the following results:

```
Volume Name         Storage       Device         Estimated      Pct     Volume
                    Pool Name     Class Name      Capacity     Util     Status
                                                      (MB)
----------------    ----------    ----------     ---------    -----    --------
/dev/vol3           BACKUPPOOL    DISK                15.0      0.0    On-Line
```

Moving Data by Node

You can move data located in a sequential-access storage pool for one or more nodes, or a single node with selected file spaces. In contrast to previous sections that discuss moving volumes using the MOVE DATA command, for the following sections you need to use the MOVE NODEDATA command. For this command the data can be located on either a primary or copy storage pool. When the source storage pool is a primary pool, you can move data to other volumes within the same pool or to another primary pool. When the source storage pool is a copy pool, data can only be moved to other volumes within that pool.

Notes:

1. You can only move data by node if the data resides in a storage pool whose data format is NATIVE or NONBLOCK.

2. If you are moving files within the same storage pool, there must be volumes available that do not contain the data you are moving. That is, the server cannot use a destination volume containing data that will need to be moved.

Task	Required Privilege Class
Move data by node	System, unrestricted storage or restricted storage

Moving Data for All File Spaces for One or More Nodes

Moving data for all file spaces on one or more nodes is useful:

- When you want to optimize performance by reducing the number of volume mounts required during a restore operation by consolidating data for a specific node or nodes within a storage pool
- When you want to move data for specified nodes into a different storage pool
- When you want to increase performance of client restore processing by first moving data to a random-access storage pool

Note: You should avoid movement of data into, out of, or within a storage pool while MOVE NODEDATA is concurrently processing data on the same storage pool.

To move all file spaces for a single node named ACCOUNTING where the data is in storage pool ACCTPOOL and the destination storage pool is BACKUPPOOL enter:

```
move nodedata accounting fromstgpool=acctpool tostgpool=backuppool
```

Moving Data for Selected File Spaces for One Node

Moving data for selected file spaces for a single node is useful:

- When you want to optimize performance by reducing the number of volume mounts required during a restore operation by consolidating data for specific file spaces within a storage pool.
- When you want to consolidate data for critical file spaces allowing restore of these files to be given higher priority during recovery situations. This would be advantageous during data recovery when it is essential to first restore only business-critical data and then restore non-business-critical data.
- When you want to move specific file spaces into a different storage pool.
- When you want to increase performance of client restore processing by first moving data to a random-access storage pool.

For example, consider moving data for a single node and restricting the data movement to files in a specific non-Unicode file space (for this example, \\eng\e$) as well as a specific Unicode file space (for this example, \\eng\d$). The node name owning the data is ENGINEERING and it currently has data stored in the ENGPOOL storage pool. After the move is complete, the data is located in the destination storage pool BACKUPPOOL. To move the data enter the following:

```
move nodedata engineering fromstgpool=engpool
 tostgpool=backuppool filespace=\\eng\e$ unifilespace=\\eng\d$
```

Another example is to move data for a single node named MARKETING from all primary sequential-access storage pools to a random-access storage pool named DISKPOOL. First obtain a list of storage pools that contain data for node MARKETING, issue either:

```
query occupancy marketing
```

or

```
SELECT * from OCCUPANCY where node_name='MARKETING';
```

For this example the list of resulting storage pool names all begin with the characters FALLPLAN. To move the data repeat the following command for every instance of FALLPLAN. The following example displays the command for FALLPLAN3:

```
move nodedata marketing fromstgpool=fallplan3
 tostgpool=diskpool
```

A final example shows moving both non-Unicode and Unicode file spaces for a node. For node NOAH move non-Unicode file space \\servtuc\d$ and Unicode file space \\tsmserv1\e$ that has a file space ID of 2 from sequential access storage pool TAPEPOOL to random access storage pool DISKPOOL.

```
move nodedata noah  fromstgpool=tapepool tostgpool=diskpool
 filespace=\\servtuc\d$ fsid=2
```

Requesting Information about the Data Movement Process

To request information on the data movement process, enter:

```
query process
```

Figure 37 shows an example of the report that you receive about the data movement process.

```
 Process     Process Description      Status
  Number
--------     --------------------     -------------------------------------------------
       3     Move Node Data           Storage Pool 3590FC, Target Pool 3590FC Files
                                       Moved: 0, Bytes Moved: 0, Unreadable Files: 0,
                                       Unreadable Bytes: 0. Current Physical File
                                       (bytes): 268,468,584

                                       Current input volume:
                                       DST308.

                                       Current output volume:
                                       DST279.
```

Figure 37. Information on the Data Movement Process

Preventing Incomplete MOVE NODEDATA Operations

There are various reasons why an incomplete MOVE NODEDATA operation can occur. The following are the most common reasons:

- Files have been marked as damaged in the source storage pool. For more information regarding how to deal with files marked as damaged see "Correcting Damaged Files" on page 537.
- Files in the source storage pool reside on volumes whose access mode is offsite, destroyed or unavailable. To complete the move operation, bring the volumes onsite, restore destroyed volumes from a copy storage pool or make the volumes available.
- Files were moved, added or deleted during the move operation. To prevent this situation, avoid the following operations during move processing:
  - Migration of any type relating to the storage pool
  - Reclamation of volumes within the storage pool
  - Simultaneously running MOVE DATA processing for a volume in a storage pool that contains data to be moved during MOVE NODEDATA processing
  - Backup operations into a copy storage pool while a MOVE NODEDATA is running for that copy pool
  - Storage of files from a client directly into the storage pool

Renaming a Storage Pool

You can rename a storage pool. You may need to do this when distributing policy using enterprise configuration. See "Setting Up a Managed Server" on page 446.

When you rename a storage pool, any administrators with restricted storage privilege for the storage pool automatically have restricted storage privilege to the storage pool under the new name. If the renamed storage pool is in a storage pool hierarchy, the hierarchy is preserved.

Copy groups and management classes may contain a storage pool name as a destination. If you rename a storage pool used as a destination, the destination in a copy group or management class is not changed to the new name of the storage pool. To continue to use the policy with the renamed storage pool as a destination, you need to change the destination in the copy groups and management classes. You then activate the policy set with the changed destinations.

Defining a Copy Storage Pool

Use a copy storage pool to back up one or more primary storage pools. See Table 20 on page 220 and "Backing Up Storage Pools" on page 507 for more information. When you define a copy storage pool, be prepared to provide some or all of the information in Table 19.

Note: To back up a primary storage pool the DATAFORMAT must be NATIVE or NONBLOCK.

Table 19. Information for Defining a Copy Storage Pool

Information	Explanation
Device class	Specifies the name of the device class assigned for the storage pool. This is a required parameter.
Pool type	Specifies that you want to define a copy storage pool. This is a required parameter. Updating a storage pool cannot change whether the pool is a primary or copy storage pool.

Table 19. Information for Defining a Copy Storage Pool (continued)

Information	Explanation
Access mode	Defines access to volumes in the storage pool for user operations (such as backup and restore) and system operations (such as reclamation). Possible values are: **Read/Write** User and system operations can read from or write to the volumes. **Read-Only** User operations can read from the volumes, but not write. However, system processes can move files within the volumes in the storage pool. **Unavailable** Specifies that users cannot access files stored on volumes in the copy storage pool. Files can be moved within the volumes of the copy storage pool, but no new writes are permitted to the volumes in the storage pool from volumes outside the storage pool.
Maximum number of scratch volumes	When you specify a value greater than zero, the server dynamically acquires scratch volumes when needed, up to this maximum number. This is a required parameter. For automated libraries, set this value equal to the physical capacity of the library. See "Maintaining a Supply of Scratch Volumes in an Automated Library" on page 127.
Collocation	When collocation is enabled, the server attempts to keep all files belonging to a client node or a client file space on a minimal number of sequential access storage volumes. See "Collocation on Copy Storage Pools" on page 188.
Reclamation threshold	Specifies when to initiate reclamation of volumes in the copy storage pool. Reclamation is a process that moves any remaining active, fragmented files from one volume to another volume, thus making the original volume available for reuse. A volume is eligible for reclamation when the percentage of unused space on the volume is greater than the reclaim parameter value. Reclamation processing works differently for offsite storage pool volumes and virtual volumes. When a copy storage pool volume that is offsite becomes eligible for reclamation, the reclamation process attempts to retrieve the active files on the reclaimable volume from a primary or copy storage pool volume that is onsite. The process then writes these files to an available volume in the original copy storage pool. See "Reclamation for Copy Storage Pools" on page 193 and "Reclamation of Volumes with the Device Type of SERVER" on page 192 for more details.
Reuse delay period	Specifies the number of days that must elapse after all of the files have been deleted from a volume before the volume can be rewritten or returned to the scratch pool. See "Delaying Reuse of Reclaimed Volumes" on page 195.

Example: Defining a Copy Storage Pool

Assume you need to maintain copies of the files stored in BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL (default disk storage pools) for disaster recovery purposes. You want to create a copy storage pool named

DISASTER-RECOVERY. You decide to use only scratch tapes in the new pool, setting the maximum number of scratch volumes to an appropriate value. You enter the following command:

```
define stgpool disaster-recovery tapeclass pooltype=copy
maxscratch=100
```

To store data in the new storage pool, you must back up the primary storage pools (BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL) to the DISASTER-RECOVERY pool. See "Backing Up Storage Pools" on page 507.

Comparing Primary and Copy Storage Pools

Table 20 compares the characteristics of primary and copy storage pools.

Table 20. Comparing Primary and Copy Storage Pools

Characteristic	Primary storage pool	Copy storage pool
Destination for backed-up or archived files (specified in backup or archive copy groups)	Yes	No
Destination for space-managed files (specified in the management class)	Yes	No
Offsite access mode for volumes	No	Yes, except for volumes with device type SERVER
Destroyed access mode for volumes	Yes	No
Random access storage volumes	Yes	No
Sequential access storage volumes	Yes	Yes
Contents	Client files (backup versions, archived files, space-managed files)	Copies of files that are stored in primary storage pools
Moving data allowed	Within the same primary storage pool, or to any primary storage pool	Within the same pool only. If volumes are offsite, data is copied from the original files in primary storage pools.
Collocation	Yes (sequential access storage pools only)	Yes
Reclamation	Yes (sequential access storage pools only)	Yes Virtual volumes (volumes with device type SERVER) and offsite volumes are handled differently. For details, see "Reclamation of Volumes with the Device Type of SERVER" on page 192 and "Reclamation of Offsite Volumes" on page 193.

Table 20. Comparing Primary and Copy Storage Pools (continued)

Characteristic	Primary storage pool	Copy storage pool
File deletion	Files are deleted: • During inventory expiration processing, if the files have expired • When a file space is deleted • When a volume is deleted with the option to discard the data • When a primary storage pool volume is audited with the FIX=YES option, if the files on the volume are damaged and no other copies of the file exist	Files are deleted: • Whenever the primary copy of the file is deleted from the primary storage pool (because of expiration, file space deletion, or volume deletion) • When a volume is deleted with the option to discard the data • When a copy storage pool volume is audited with the FIX=YES option, if the files on the volume are damaged

Deleting a Storage Pool

Task	Required Privilege Class
Delete storage pools	System

Before you delete a storage pool, ensure that:

- All volumes within the storage pool have been deleted

  Ensure that you have saved any readable data that you want to preserve by issuing the MOVE DATA command. Moving all of the data that you want to preserve may require you to issue the MOVE DATA command several times.

  Before you begin deleting all volumes that belong to the storage pool, change the access mode of the storage pool to unavailable so that no files can be written to or read from volumes in the storage pool.

  See “Deleting a Storage Pool Volume with Data” on page 222 for information about deleting volumes.
- The storage pool is not identified as the next storage pool within the storage hierarchy

  To determine whether this storage pool is referenced as the next storage pool within the storage hierarchy, query for storage pool information as described in “Monitoring Space Available in a Storage Pool” on page 198.

  Update any storage pool definitions to remove this storage pool from the storage hierarchy by performing one of the following:
  – Naming another storage pool as the next storage pool in the storage hierarchy
  – Entering the value for the NEXTSTGPOOL parameter as "" (double quotes) to remove this storage pool from the storage hierarchy definition

  See “Defining or Updating Primary Storage Pools” on page 158 for information about defining and updating storage pools.
- The storage pool to be deleted is not specified as the destination for any copy group in any management class within the active policy set of any domain. Also, a storage pool to be deleted cannot be the destination for space-managed files (specified in any management class within the active policy set of any domain). If this pool is a destination and the pool is deleted, operations fail because there is no storage space to store the data.

Deleting Storage Pool Volumes

You can delete volumes, and optionally the client files they contain, from either primary or copy storage pools.

If files that are not cached are deleted from a primary storage pool volume, any copies of these files in copy storage pools will also be deleted.

Files in a copy storage pool are never deleted unless:

- The volume that contains the copy file is deleted by using the DISCARDDATA=YES option.
- A read error is detected by using AUDIT VOLUME with the FIX=YES option for a copy storage pool volume.
- The primary file is deleted because of:
  - Policy-based file expiration
  - File space deletion
  - Deletion of the primary storage pool volume

Tip: If you are deleting many volumes, delete the volumes one at a time. Concurrently deleting many volumes can adversely affect server performance.

Task	**Required Privilege Class**
Delete volumes from any storage pool	System or unrestricted storage
Delete volumes from storage pools over which they have authority	Restricted storage

Deleting an Empty Storage Pool Volume

You can delete empty storage pool volumes. For example, to delete an empty volume named WREN03, enter:

```
delete volume wren03
```

On an administrative client, you will receive the following confirmation messages, unless the client is running with the NOCONFIRM option:

```
ANR2200W  This command will delete volume WREN03
from its storage pool after verifying that the volume
contains no data.
Do you wish to proceed? (Y/N)
```

After you respond yes, the server generates a background process to delete the volume.

The command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

Deleting a Storage Pool Volume with Data

To prevent you from accidentally deleting backed-up, archived, or space-managed files, the server does not allow you to delete a volume that contains user data unless you specify DISCARDDATA=YES on the DELETE VOLUME command.

For example, to discard all data from volume WREN03 and delete the volume from its storage pool, enter:

```
delete volume wren03 discarddata=yes
```

The server generates a background process and deletes data in a series of batch database transactions. After all files have been deleted from the volume, the server deletes the volume from the storage pool. If the volume deletion process is canceled or if a system failure occurs, the volume might still contain data. Reissue the DELETE VOLUME command and explicitly request the server to discard the remaining files on the volume.

To delete a volume but not the files it contains, move the files to another volume. See "Moving Files from One Volume to Another Volume" on page 211 for information about moving data from one volume to another volume.

Residual data: Even after you move data, residual data may remain on the volume because of I/O errors or because of files that were previously marked as damaged. (TSM does not move files that are marked as damaged.) To delete any volume that contains residual data that cannot be moved, you must explicitly specify that files should be discarded from the volume.

Part 3. Managing Client Operations

© Copyright IBM Corp. 1993, 2002

Chapter 10. Adding Client Nodes

When the Tivoli Storage Manager server is installed, the Tivoli Storage Manager backup-archive client and the administrative client are installed on the same machine as the server by default. However, many installations of Tivoli Storage Manager include remote clients, and application clients on other machines, often running on different operating systems.

The server views its registered clients as nodes that require services and resources from the server. The term *nodes* in this chapter indicates the following type of clients and servers that you can register as client nodes:

- Tivoli Storage Manager backup-archive client
- Tivoli Data Protection application clients
- Tivoli Space Manager (HSM client)
- Tivoli Storage Manager source server registered as a node on a target server
- Network-attached storage (NAS) file server using Tivoli Data Protection for NDMP support

Each node must be registered with the server and requires an option file with a pointer to the server.

For details on many of the topics in this chapter, refer to *Backup-Archive Installation and User's Guide*. Administrators can perform the following activities when managing nodes:

Tasks:
"Installing Client Node Software" on page 228
"Accepting Default Closed Registration or Enabling Open Registration" on page 228
"Registering Nodes with the Server" on page 228
"Connecting Nodes with the Server" on page 231
Concepts:
"Overview of Clients and Servers as Nodes" on page 228
"Comparing Network-Attached Nodes to Local Nodes" on page 233

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

© Copyright IBM Corp. 1993, 2002

Overview of Clients and Servers as Nodes

Each backup-archive client, HSM client, TDP application client, and source server is given a node name when it is registered as a node with the Tivoli Storage Manager server. The server considers each as a node that requires services and resources from the server.

Typically, a node is equivalent to a machine as in the case of a backup-archive client that is installed on a user's computer for file system backups. However, multiple nodes can exist on a single machine. For example, a Structured Query Language (SQL) server machine can contain both a TDP for SQL server application client for database and transaction log backups, and a Tivoli Storage Manager backup-archive client for file system backups.

Installing Client Node Software

Administrators can install backup-archive clients, application clients, or Tivoli Space Manager clients by using any of the following methods:

- Installing directly from the CD-ROM
- Installing by transferring installable files from the CD-ROM to a target machine
- Installing by creating client software images and installing the images

For more information about installing:

- Client software, refer to *Backup-Archive Installation and User's Guide*.
- Tivoli Data Protection application client software, refer to the Tivoli Data Protection application client documentation for your particular client.

Use the procedures in this chapter to configure a node after it has been installed.

Registering Nodes with the Server

Administrators can register Tivoli Storage Manager clients, Tivoli Data Protection application clients, and HSM clients as client nodes.

When a node is registered, Tivoli Storage Manager automatically creates an administrative user ID with client owner authority over the node. You can use this administrative user ID to access the Web backup-archive client from remote locations through a Web browser. If an administrative user ID already exists with the same name, an administrative user ID is not automatically defined. For more information, see "Overview of Remote Access to Web Backup-Archive Clients" on page 238.

Note: To connect to a Web backup-archive client directly from a supported Web browser or from a hyperlink in the Web administrative Enterprise Console, you must specify the node's URL and port number during the registration process or later update the node with this information.

Accepting Default Closed Registration or Enabling Open Registration

Before a user can request Tivoli Storage Manager services, the node must be registered with the server.

Closed registration is the default. The administrator must register client nodes when registration is set to closed.

Open registration allows the client nodes to register their node names, passwords, and compression options. On UNIX systems, only the root user can register a client node with the server.

With either registration mode, by default, an administrative user ID with client owner authority is created over the node.

Note: Changes to the registration process do not affect existing registered client nodes.

Closed Registration

To add a node with closed registration, an administrator uses the REGISTER NODE command to register the node and specify the initial password. The administrator can also specify the following optional parameters:

- Contact information.
- The name of the policy domain to which the node is assigned.
- Whether the node compresses its files before sending them to the server for backup and archive.
- Whether the node can delete backups and archives from server storage.
- The name of a client option set to be used by the node.
- Whether to force a node to change or reset the password.
- Whether the client node keeps a mount point for an entire session.
- The maximum number of mount points the node can use.

Open Registration

To add a node with open registration, the server prompts the user for a node name, password, and contact information the first time the user attempts to connect to the server. With open registration, the server automatically assigns the node to the STANDARD policy domain. The server by default allows users to delete archive copies, but not backups stored in server storage.

You can enable open registration by entering the following command from an administrative client command line:

```
set registration open
```

For examples and a list of open registration defaults, refer to the *Administrator's Reference*.

To change the defaults for a registered node, use the UPDATE NODE command.

Node Compression Considerations

When you enable compression, it reduces network utilization and saves server storage, but causes additional central processing unit (CPU) overhead to the node. Data compression is recommended only when there is insufficient network capacity.

Attention: Use either client compression or drive compression, but not both. For details, see "Using Data Compression" on page 152.

To optimize performance or to ease memory constraints at the workstation, an administrator can restrict file compression. You can select one of three options:

- Compress files
- Do not compress files
- Use the value set in the COMPRESSION option

Set the COMPRESSION option in the client system options file or in the application program interface (API) configuration file.

On a UNIX system, a root user can define the COMPRESSION option in the dsm.opt client options file.

Registering Nodes with Client Options Sets

Administrators can use client options sets in conjunction with the client options file to register nodes with the server. Client option sets are considered advanced implementation and are discussed in "Modifying Client Option Files" on page 253. You can specify an option set for a node when you register or update the node. For example:

```
register node mike pass2eng cloptset=engbackup
```

The client node MIKE is registered with the password pass2eng. When the client node MIKE performs a scheduling operation, the schedule log entries are kept for 5 days.

Registering a Network-attached Storage File Server as a Node

To include a NAS file server as a node that TSM can back up and restore with Tivoli Data Protection for NDMP support, you can register the file server as a NAS node. Data that is backed up from the NAS file server will be associated with the NAS node name.

The REGISTER NODE and UPDATE NODE commands have a default parameter of TYPE=CLIENT. To register a NAS file server as a node, you must specify the TYPE=NAS parameter. For example to register a NAS file server with a node name of NASXYZ and a password of PW4PW, enter the following:

```
register node nasxyz pw4pw type=nas
```

You must use this same node name when you later define the corresponding data mover name. For more information, see "Chapter 6. Setting Up Tivoli Data Protection for NDMP" on page 97.

Registering a Source Server as a Node on a Target Server

A virtual volume is a volume that appears to be a sequential media volume on a source server. The volume is actually stored as an archive file on a target server.

To use virtual volumes, register the source server as a client node on the target server.

The REGISTER NODE and UPDATE NODE commands have a default parameter of TYPE=CLIENT. To register a source server as a node, you must specify the TYPE=SERVER parameter. For more information, see "Using Virtual Volumes to Store Data on Another Server" on page 469.

Registering an Application Programming Interface to the Server

Workstation users can request TSM services by using an application that uses the TSM application programming interface (API). An administrator uses the REGISTER NODE command to register the workstation as a node.

Understanding How to Set the Compression Option

For applications that use the TSM API, compression can be determined by:

- An administrator during registration who can:
  - Require that files are compressed
  - Restrict the client from compressing files
  - Allow the application user or the client user to determine the compression status
- The client options file. If an administrator does not set compression on or off, Tivoli Storage Manager checks the compression status that is set in the client options file. The client options file is required, but the API user configuration file is optional.
- One of the object attributes. When an application sends an object to the server, some object attributes can be specified. One of the object attributes is a flag that indicates whether or not the data has already been compressed. If the application turns this flag on during either a backup or an archive operation, then Tivoli Storage Manager does not compress the data a second time. This process overrides what the administrator sets during registration.

For more information on setting options for the API and on controlling compression, see *Tivoli Storage Manager Using the Application Program Interface*.

Understanding How to Set the File Deletion Option

For applications that use the TSM API, the file deletion option can be set by:

- An administrator during registration

  If an administrator does not allow file deletion, then an administrator must delete objects or file spaces that are associated with the workstation from server storage.

  If an administrator allows file deletion, then Tivoli Storage Manager checks the client options file.
- An application using the TSM API deletion program calls

  If the application uses the **dsmDeleteObj** or **dsmDeleteFS** program call, then objects or files are marked for deletion when the application is executed.

Connecting Nodes with the Server

The client options file connects each node to the server. Administrators and users on all platforms can modify their client options file (*dsm.opt*) with a text editor. Client options files can be updated differently across platforms. On the Windows platform, you can use a wizard to work with the client options file.

Note: If any changes are made to the dsm.opt file, the client must be restarted for changes in the options file to have any affect.

The client options file *dsm.opt* is located in the client, application client, or host server directory. If the file does not exist, copy the dsm.smp file. Users and administrators can edit the client options file to specify:

- The network address of the server
- The communication protocol
- Backup and archive options
- Space management options
- Scheduling options

Required Client Options

Each node requires a client options file. Each client options file must contain the network address of the Tivoli Storage Manager server and other communication options that allow the node to communicate with the server. Figure 38 shows the contents of a client options file that is configured to connect to the server by using TCP/IP. The communication options specified in the client options file satisfy the minimum requirements for the node to connect with the server.

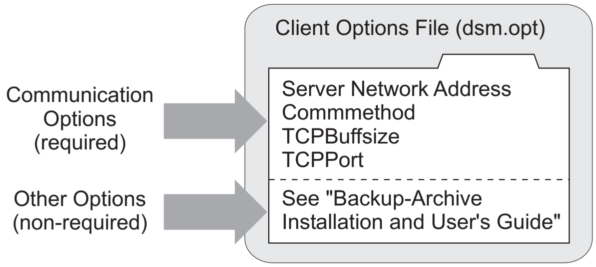

Client Options File (dsm.opt)
Communication Options (required)
Other Options (non-required)
Server Network Address
Commmethod
TCPBuffsize
TCPPort
See "Backup-Archive Installation and User's Guide"

Figure 38. Client Node Options File

Non–Required Client Options

Many non-required options are available that can be set at any time. These options control the behavior of Tivoli Storage Manager processing. Refer to *Backup-Archive Installation and User's Guide* for more information about non-required client options.

UNIX Client Options

For UNIX clients, options are located in three options files: client systems options file, client user options file, and include-exclude options file. Clients on other platforms use a single options file.

Methods for Creating or Updating a Client Options File

There are several methods for creating or updating client options files. The available methods depend on the client platform.

Using a Text Editor

All options files (*dsm.opt*) can be edited with a text editor. Anyone can edit the client options file if they have access to the directory where the node software is installed. Editing individual options files is the most direct method, but may not be suitable for sites with many client nodes.

Using the Client Configuration Wizard

When a local backup-archive client GUI starts initially and TSM does not find an options file, a setup wizard guides the client through the configuration process.

From the backup-archive client GUI, the client can also display the setup wizard by selecting **Utilities→Setup Wizard**. The client can follow the panels in the setup wizard to browse TSM server information in the Active Directory. The client can determine which server to connect to and what communication protocol to use.

Note: This wizard is not available for the Web client.

Comparing Network-Attached Nodes to Local Nodes

A Tivoli Storage Manager environment can be either a server and client on the same machine (stand-alone environment) or a server and network-attached clients (network environment).

The stand-alone environment of Tivoli Storage Manager consists of a backup-archive client and a administrative client on the same computer as the server. There is nothing more to do to connect the client. This is shown in Figure 39.

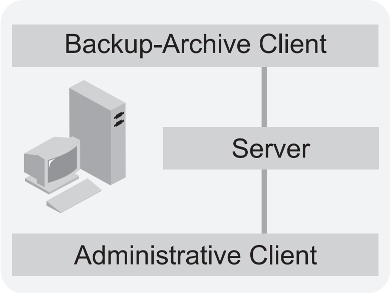


Figure 39. Stand-alone Environment

Figure 40 shows that a network environment of Tivoli Storage Manager consists of a backup-archive client and an administrative client on the same computer as the server. However, network-attached client nodes can also connect to the server.

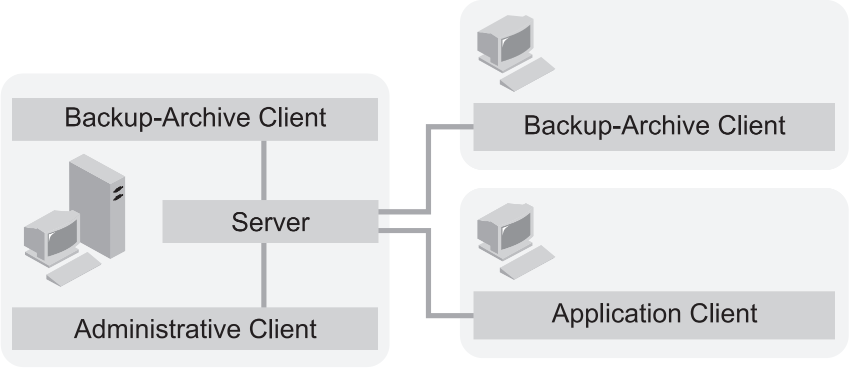


Figure 40. Network Environment

Each client requires a client options file. A user can edit the client options file at the client node. The options file contains a default set of processing options that identify the server, communication method, backup and archive options, space management options, and scheduling options.

Adding Clients from the Administrative Command Line Client

The administrator can register nodes by using the REGISTER NODE command. For more information, refer to *Administrator's Reference*.

Enabling Open Registration

The default registration mode at installation is closed. To change the default to open so users an register their own client nodes, enter:

```
set registration open
```

Configuring the Client Options File to Connect with the Server

Edit the client options file (dsm.opt) in the client directory using a text editor.

Example: Register Three Client Nodes Using the Administrative Command Line

You want to register three workstations from the engineering department and assign them to the ENGPOLDOM policy domain. Before you can assign client nodes to a policy domain, the policy domain must exist. To define a policy domain, see "Chapter 12. Implementing Policies for Client Data" on page 269.

You want to let users delete backed up or archived files from storage pools. From an administrative client, you can use the macro facility to register more than one client node at a time. For this example, you create a macro file named REGENG.MAC, that contains the following REGISTER NODE commands:

```
register node ssteiner choir contact='department 21'
domain=engpoldom archdelete=yes backdelete=yes

register node carolh skiing contact='department 21, second shift'
domain=engpoldom archdelete=yes backdelete=yes

register node mab guitar contact='department 21, third shift'
domain=engpoldom archdelete=yes backdelete=yes
```

Next, issue the MACRO command:

```
macro regeng.mac
```

For information on the MACRO command, see *Administrator's Reference*.

Chapter 11. Managing Client Nodes

This chapter contains information about managing client nodes that have been installed and configured. For information about installing and configuring client nodes, see “Chapter 10. Adding Client Nodes” on page 227.

Note: The Tivoli Storage Manager server views its registered clients, application clients, and source servers as nodes. In a general sense, the server considers all of these entities as a client. The term *nodes* in this chapter refers to the following type of clients and servers as client nodes:

- Tivoli Data Protection application clients
- Tivoli Storage Manager backup-archive client
- Tivoli Storage Manager source server registered as a node on a target server
- Network-attached storage (NAS) file server using Tivoli Data Protection for NDMP support

Administrators can manage client nodes and control their access to the server. See the following sections for more information:

Tasks:
“Managing Nodes” on page 236
“Managing Client Access Authority Levels” on page 240
“Managing File Spaces” on page 242
“Modifying Client Option Files” on page 253
“Managing Tivoli Storage Manager Sessions” on page 256
“Managing Tivoli Storage Manager Security” on page 260
Concepts:
“Overview of Client Nodes and File Spaces” on page 242

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator’s Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Managing Client Node Registration Techniques

By default, Tivoli Storage Manager provides closed registration as the technique for registering client nodes. Administrators can modify the default with the SET REGISTRATION command. For more information about open and closed registration, see “Accepting Default Closed Registration or Enabling Open Registration” on page 228.

© Copyright IBM Corp. 1993, 2002

Managing Nodes

From the perspective of the server, each client and application client is a node requiring Tivoli Storage Manager services. For information, see “Overview of Client Nodes and File Spaces” on page 242. Client nodes can be local or remote to the server. For information, see “Comparing Network-Attached Nodes to Local Nodes” on page 233.

Administrators can perform the following activities when managing client nodes.

Task	Required Privilege Class
Updating, renaming, locking, or unlocking any client nodes	System or unrestricted policy
Updating, renaming, locking, or unlocking client nodes assigned to specific policy domains	System, unrestricted policy, or restricted policy for those domains
Displaying information about client nodes or file spaces	Any administrator
Deleting any client nodes	System or unrestricted policy
Removing client nodes assigned to specific policy domains	System, unrestricted policy, or restricted policy for those domains
Managing client access authority levels	System

Updating Client Node Information

You can use the UPDATE NODE command to update information such as the client’s assigned policy domain, the user’s password or contact information, and the client option set used by the node.

For example, update client node TOMC to prevent him from deleting archived files from storage pools by entering:

```
update node tomc archdelete=no
```

Renaming Client Nodes

You can rename a client node with the RENAME NODE command. You may need to rename a client node if the workstation network name or host name changes. For example, with UNIX clients, users define their node name based on the value returned by the HOSTNAME command. When users access the server, their Tivoli Storage Manager user IDs match the host name of their workstations. If the host name changes, you can update a client node user ID to match the new host name.

For example, to rename CAROLH to ENGNODE, enter:

```
rename node carolh engnode
```

ENGNODE retains the contact information and access to backup and archive data that belonged to CAROLH. All files backed up or archived by CAROLH now belong to ENGNODE.

Locking and Unlocking Client Nodes

You can prevent client nodes from accessing the server with the LOCK NODE command. This will prevent client nodes from performing functions such as either backup and restore or archive and retrieve.

You can restore a locked node’s access to the server with the UNLOCK NODE command.

For example, to prevent client node MAB from accessing the server, enter:

```
lock node mab
```

To let client node MAB access the server again, enter:

```
unlock node mab
```

Deleting Client Nodes

You can delete a client node from the server with the REMOVE NODE command. All file spaces that belong to the client node must first be deleted from server storage. After all of the client node’s file spaces have been deleted (see “Deleting File Spaces and Client Nodes” on page 252), you can delete the node.

For example, to remove client node DEBBYG, enter:

1. Delete the DEBBYG file space by entering:

   ```
   delete filespace debbyg * type=any
   ```

2. Delete the DEBBYG node by entering:

   ```
   remove node debbyg
   ```

Note: Before you can delete a NAS node, you must first delete any file spaces, then delete any defined paths for the data mover with the DELETE PATH command. Delete the corresponding data mover with the DELETE DATAMOVER command. Then you can issue the REMOVE NODE command to delete the NAS node.

Displaying Information about Client Nodes

You can display information about client nodes. For example, as a policy administrator, you might query the server about all client nodes assigned to the policy domains for which you have authority. Or you might query the server for detailed information about one client node.

Displaying Information about Client Nodes Assigned to Specific Policy Domains

You can display information about client nodes assigned to specific policy domains. For example, to view information about client nodes that are assigned to STANDARD and ENGPOLDOM policy domains, enter:

```
query node * domain=standard,engpoldom
```

The output from that command may display similar to the following:

```
Node Name    Platform     Policy Domain   Days Since   Days Since   Locked?
                          Name                  Last     Password
                                              Access          Set
----------   --------     --------------  ----------   ----------   -------
DEBBYG         DOS        STANDARD                 2           12   No
ENGNODE        AIX        ENGPOLDOM               <1            1   No
HTANG          OS/2       STANDARD                 4           11   No
MAB            AIX        ENGPOLDOM               <1            1   No
PEASE          AIX        STANDARD                 3           12   No
SSTEINER       (?)        ENGPOLDOM               <1            1   No
```

Displaying Information about a Specific Client Node

You can view information about specific client nodes. For example, to review the registration parameters defined for client node JOE, enter:

```
query node joe format=detailed
```

The resulting report may appear similar to the following:

```
                  Node Name: JOE
                   Platform: WinNT
            Client OS Level: 4.00
             Client Version: Version 3, Release 1, Level 3.0
         Policy Domain Name: STANDARD
      Last Access Date/Time: 05/19/1999 18:55:46
     Days Since Last Access: 6
     Password Set Date/Time: 05/19/1999 18:26:43
    Days Since Password Set: 6
      Invalid Sign-on Count: 0
                    Locked?: No
                    Contact:
                Compression: Client's Choice
    Archive Delete Allowed?: Yes
     Backup Delete Allowed?: No
     Registration Date/Time: 05/19/1999 18:26:43
  Registering Administrator: SERVER_CONSOLE
Last Communication Method Used: Tcp/Ip
   Bytes Received Last Session: 108,731
       Bytes Sent Last Session: 698
Duration of Last Session (sec): 0.00
   Pct. Idle Wait Last Session: 0.00
  Pct. Comm. Wait Last Session: 0.00
  Pct. Media Wait Last Session: 0.00
                    Optionset:
                          URL:http://joe.host.name:1581
                    Node Type: Client
    Password Expiration Period: 60
             Keep Mount Point?: No
  Maximum Mount Points Allowed: 1
        Auto Filespace Rename: No
            Validate Protocol: No
```

Overview of Remote Access to Web Backup-Archive Clients

With the introduction of the Web backup-archive client, when a client node is registered with a Tivoli Storage Manager 3.7.0 server or above, an identical administrative user ID is created at the same time. This user ID has client owner authority over the node by default.

Enterprise logon enables a user with the proper administrative user ID and password to access a Web backup-archive client from a Web browser. The Web backup-archive client can be used by the client node or a user ID with the proper authority to perform backup, archive, restore, and retrieve operations on any machine that is running the Web backup-archive client.

You can establish access to a Web backup-archive client for help desk personnel that do not have system or policy privileges by granting those users client access authority to the nodes they need to manage. Help desk personnel can then perform activities on behalf of the client node such as backup and restore operations.

A native backup-archive client can log on to Tivoli Storage Manager using their node name and password, or administrative user ID and password. The administrative user ID password is managed independently from the password that is generated with the ***passwordaccess generate*** client option. The client must have the option ***passwordaccess generate*** specified in their client option file to enable use of the Web backup-archive client.

To use the Web backup-archive client from your web browser, you specify the URL and port number of the Tivoli Storage Manager backup-archive client machine running the Web client. The browser you use to connect to a Web backup-archive client must be Microsoft® Internet Explorer 5.0 or Netscape 4.7 or later. The browser must have the Java Runtime Environment (JRE) 1.3.1, which includes the Java Plug-in software. The JRE is available at the following URL, http://java.sun.com/getjava.

During node registration, you have the option of granting client owner or client access authority to an existing administrative user ID. You can also prevent the server from creating an administrative user ID at registration. If an administrative user ID already exists with the same name as the node being registered, the server registers the node but does not automatically create an administrative user ID. This process also applies if your site uses open registration.

For more information about installing and configuring the Web backup-archive client, refer to *Backup-Archive Installation and User's Guide*.

Description of Node Privilege Class with Client Access Authorities

Access to a Web backup-archive client requires either client *owner* authority or client *access* authority. Administrators with system or policy privileges over the client node's domain, have client owner authority by default. The administrative user ID created automatically at registration has *client owner* authority by default. This administrative user ID is displayed when an administrator issues a QUERY ADMIN command.

The following describes the difference between client *owner* and client *access* authority when defined for a user that has the node privilege class:

Client owner

You can access the client through the Web backup-archive client or native backup-archive client.

You own the data and have a right to physically gain access to the data remotely. You can backup and restore files on the same or different machine, you can delete file spaces or archive data.

The user ID with client owner authority can also access the data from another machine using the –NODENAME parameter.

The administrator can change the client node's password for which they have authority.

This is the default authority level for the client at registration. An administrator with system or policy privileges to a client's domain has client owner authority by default.

Client access

You can only access the client through the Web backup-archive client.

You can restore data only to the original client.

A user ID with client access authority cannot access the client from another machine using the –NODENAME parameter.

This privilege class authority is useful for help desk personnel so they can assist users in backing up or restoring data without having system or policy privileges. The client data can only be restored to none other than the original client. A user ID with client access privilege cannot directly access client's data from a native backup-archive client.

Managing Client Access Authority Levels

By default, an administrator with system or policy privilege over a client's domain can remotely access clients and perform backup and restore operations.

You can grant client *access* or client *owner* authority to other administrators by specifying CLASS=NODE and AUTHORITY=ACCESS or AUTHORITY=OWNER parameters on the GRANT AUTHORITY command. You must have one of the following privileges to grant or revoke client access or client owner authority:

- System privilege
- Policy privilege in the client's domain
- Client owner privilege over the node
- Client access privilege over the node

You can grant an administrator client access authority to individual clients or to all clients in a specified policy domain. For example, you may want to grant client access privileges to users that staff help desk environments. See “Example: Setting up Help Desk Access to Client Machines in a Specific Policy Domain” on page 241 for more information.

Granting Client Authority

To grant client *access* authority to administrator FRED for the LABCLIENT node, issue:

```
grant authority fred class=node node=labclient
```

The administrator FRED can now access the LABCLIENT client, and perform backup and restore. The administrator can only restore data to the LABCLIENT node.

To grant client *owner* authority to ADMIN1 for the STUDENT1 node, issue:

```
grant authority admin1 class=node authority=owner node=student1
```

The user ID ADMIN1 can now perform backup and restore operations for the STUDENT1 client node. The user ID ADMIN1 can also restore files from the STUDENT1 client node to a different client node.

Automatically Creating an Administrative User ID with Client Owner Authority

When you use the REGISTER NODE command, by default, the server creates an administrative user ID in addition to the client node. The administrative user ID has client owner authority to the node when the node is defined to the server. For example, you want to register client node DESK2, issue:

```
 register node desk2 pass2dsk
```

The following shows the output from this command.

```
ANR2060I Node DESK2 registered in policy domain STANDARD.
ANR2099I Administrative userid DESK2 defined for OWNER access to node DESK2.
```

The DESK2 client node is registered, in addition to an administrative user ID with the same ID. The administrative user ID DESK2 has a password of pass2dsk with client owner authority to the DESK2 node. When the PASSWORDACCESS=GENERATE option is used by the client to change the password, the administrative DESK2 ID can still access the client from a remote location.

Preventing Automatic Creation of an Administrative User ID with Client Owner Authority

You can prevent automatic creation of an administrative user ID with client owner authority by specifying USERID=NONE on the REGISTER NODE command. For example, you want to register DESK2 without creating an administrative user ID with client owner authority by default. Issue the following:

```
register node desk2 pass2dsk userid=none
```

Registering a Node and Granting an Existing Administrative ID Client Owner Authority

You can grant client owner authority to an existing administrative user ID. For example, to give client owner authority to the HELPADMIN user ID when registering the NEWCLIENT node, enter:

```
register node newclient pass2new userid=helpadmin
```

This command results in the NEWCLIENT node being registered with a password of pass2new, and also grants HELPADMIN client owner authority. This command would not create an administrator ID. The HELPADMIN client user ID is now able to access the NEWCLIENT node from a remote location.

Example: Setting up Help Desk Access to Client Machines in a Specific Policy Domain

You want to set up help desk access for user HELP1 to the client nodes in the FINANCE domain. You want to grant HELP1 client access authority to the FINANCE domain without having to grant system or policy privileges.

The client nodes have been previously set up as follows:

- Installed and configured. The URL and port numbers were specified during the REGISTER NODE process.
- Assigned to the FINANCE policy domain.
- Started the TSM Client Acceptor service.
- Specified ***passwordaccess generate*** option in their client option files.

The help desk person, using HELP1 user ID, has a Web browser with Java Runtime Environment (JRE) 1.3.1.

1. Register an administrative user ID of HELP1.

   ```
   register admin help1 05x23 contact="M. Smith, Help Desk x0001"
   ```

2. Grant the HELP1 administrative user ID client access authority to all clients in the FINANCE domain. With client access authority, HELP1 can perform backup and restore operations for clients in the FINANCE domain. Client nodes in the FINANCE domain are Dave, Sara, and Joe.

   ```
   grant authority help1 class=node authority=access domains=finance
   ```

   The following is output generated by this command:

```
ANR2126I GRANT AUTHORITY: Administrator HELP1 was granted ACCESS authority for client
                          DAVE.
ANR2126I GRANT AUTHORITY: Administrator HELP1 was granted ACCESS authority for client
                          JOE.
ANR2126I GRANT AUTHORITY: Administrator HELP1 was granted ACCESS authority for client
                          SARA.
```

3. The help desk person, HELP1, opens the Web browser and specifies the URL and port number for client machine Sara:

   ```
   http://sara.machine.name:1581
   ```

   A Java applet is started, and the client hub window is displayed in the main window of the Web browser. When HELP1 accesses the backup function from the client hub, the Tivoli Storage Manager login screen is displayed in a separate Java applet window. HELP1 authenticates with the administrative user ID and password. HELP1 can perform a backup for Sara.

For information about what functions are not supported on the Web backup-archive client, refer to *Backup-Archive Installation and User's Guide*.

Managing File Spaces

A *file space name* identifies a group of files that are stored as a logical unit in server storage. Administrators manage file spaces in which Tivoli Storage Manager stores each client node's data. See "Overview of Client Nodes and File Spaces" for more information.

Administrators can perform the following activities when managing file spaces:

Task	**Required Privilege Class**
Determine when existing file spaces are renamed to allow for the creation of new Unicode-enabled file spaces	System, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node is assigned.
Displaying information about file spaces	Any administrator
Move selected file spaces for a single node, as well as move a node's data located in a sequential access storage pool	System, unrestricted storage, or restricted storage privilege for the source storage pool. If your authorization is restricted storage privilege and you intend to move data to another storage pool, you must also have the appropriate authority for the destination storage pool.
Deleting file spaces	System or unrestricted policy
Deleting file spaces assigned to specific policy domains	System, unrestricted policy, or restricted policy for those domains

Overview of Client Nodes and File Spaces

Each client is given a node name when it is registered with the server. The server views its registered nodes as clients that require services and resources from the server.

Typically, a node is equivalent to a machine as in the case of a backup-archive client installed on a user's computer for file system backups. However, multiple nodes can exist on a single machine as in the case of a SQL server machine containing both an application client for SQL database and transaction log backups, and a backup-archive client for file system backups.

Typically, each client file system is represented on the server as a unique file space that belongs to each client node. Therefore, the number of file spaces a node has depends on the number of file systems on the client machine. For example, a Windows desktop system may have multiple drives (file systems), such as C: and D:. In this case, the client's node has two file spaces on the server; one for the C: drive and a second for the D: drive. The file spaces can grow as a client stores more data on the server. The file spaces decrease as backup and archive file versions expire and the server reclaims the space. Tivoli Storage Manager does not allow an administrator to delete a node unless the node's file spaces have been deleted.

File Spaces for Clients

For client nodes running on Windows, file spaces map to logical partitions and shares. Each file space is named with the UNC name of the respective client partition or share.

For client nodes running on NetWare, file spaces map to NetWare volumes. Each file space is named with the corresponding NetWare volume name.

For clients running on Macintosh, file spaces map to Macintosh volumes. Each file space is named with the corresponding Macintosh volume name.

For clients running on UNIX, a file space name maps to a file space in storage that has the same name as the file system or virtual mount point from which the files originated. The VIRTUALMOINTPOINT option allows users to define a virtual mount point for a file system to back up or archive files beginning with a specific directory or subdirectory. For information on the VIRTUALMOUNTPOINT option, refer to the appropriate *Backup-Archive Installation and User's Guide*.

Supporting Unicode-Enabled Clients

Unicode is a universal character encoding standard that supports the interchange, processing, and display of text that is written in any of the languages of the modern world. For Windows NT, Windows 2000, Windows 2002, Windows .NET, and Windows XP systems with the Unicode-enabled client, the server supports storing file spaces with Unicode file space names, directory names, and file names in server storage. The file spaces in server storage that have Unicode names are called *Unicode-enabled file spaces*. Support for Unicode names enables a client to successfully process a Tivoli Storage Manager operation even when the file spaces contain directory names or files in multiple languages, or when the client uses a different code page than the server.

New clients storing data on the server for the first time require no special set-up. If the client has the latest Tivoli Storage Manager client software installed, the server automatically stores Unicode-enabled file spaces for that client.

However, if you have clients that already have data stored on the server and the clients install the Unicode-enabled Tivoli Storage Manager client software, you need to plan for the migration to Unicode-enabled file spaces. To allow clients with existing data to begin to store data in Unicode-enabled file spaces, Tivoli Storage Manager provides a function for automatic renaming of existing file spaces. The file data itself is not affected; only the file space name is changed. Once the existing file space is renamed, the operation creates a new file space that is Unicode enabled. The creation of the new Unicode-enabled file space for clients can greatly increase the amount of space required for storage pools and the amount of space required for the server database. It can also increase the amount of time required for a client to run a full incremental backup, because the first incremental backup after the creation of the Unicode-enabled file space is a full backup.

When clients with existing file spaces migrate to Unicode-enabled file spaces, you need to ensure that sufficient storage space for the server database and storage pools is available. You also need to allow for potentially longer backup windows for the complete backups.

Note: Once the server is at the latest level of software that includes support for Unicode-enabled file spaces, you can only go back to a previous level of the server by restoring an earlier version of Tivoli Storage Manager and the database.

A Unicode-enabled Tivoli Storage Manager client is currently available only on Windows NT, Windows 2000, Windows 2002, Windows .NET, and Windows XP. Data in a Unicode code page from any other source, including down-level clients and API clients, will not be identified or treated as Unicode enabled.

Note: The remainder of this section will refer to these clients as Unicode-enabled clients, users of Windows NT-based operating systems, or clients.

It is strongly recommended that users of Windows NT-based operating systems migrate their non-Unicode file spaces to Unicode enabled file spaces. For more information see *Backup-Archive Installation and User's Guide*.

See the following sections:

“Reasons for Migrating Clients to Unicode-Enabled File Spaces”

“Migrating Clients to Unicode-Enabled File Spaces” on page 245

“Querying Unicode-enabled File Spaces” on page 251

“Unicode-enabled Clients and Existing Backup Sets” on page 251

Reasons for Migrating Clients to Unicode-Enabled File Spaces

Without Tivoli Storage Manager support for storing Unicode-enabled file spaces, some clients have experienced backup failures when file spaces contain names of directories or files in multiple languages, or have names that cannot be converted to the server's code page. When Tivoli Storage Manager cannot convert the code page, the client may receive one or all of the following messages if they were using the command line: ANS1228E, ANS4042E, and ANS1803E. Clients that are using the GUI may see a “Path not found” message. If you have clients that are experiencing such backup failures, then you need to migrate the file spaces for these clients to ensure that these systems are completely protected with backups. If you have a large number of clients, set the priority for migrating the clients based on how critical each client's data is to your business. See “Migrating Clients to Unicode-Enabled File Spaces” on page 245.

Any new file spaces that are backed up from client systems with the Unicode-enabled Tivoli Storage Manager client are automatically stored as Unicode-enabled file spaces in server storage.

Objects backed up or archived with a Unicode-enabled Tivoli Storage Manager client in any supported language environment can be restored or retrieved with a Unicode-enabled client in the same or any other supported language environment.

This means, for example, that files backed up by a Japanese Unicode-enabled client can be restored by a German Unicode-enabled client.

Note: Objects backed up or archived by a Unicode-enabled Tivoli Storage Manager client, cannot be restored or retrieved by a client that is not Unicode enabled.

Migrating Clients to Unicode-Enabled File Spaces

To allow clients with existing data to migrate to Unicode-enabled file spaces, Tivoli Storage Manager provides an automatic rename function for file spaces. When enabled, Tivoli Storage Manager uses the rename function when it recognizes that a file space that is not Unicode enabled in server storage matches the name of a file space on a client. The existing file space in server storage is renamed, so that the file space in the current operation is then treated as a new, Unicode-enabled file space. For example, if the operation is an incremental backup at the file space level, the entire file space is then backed up to the server as a Unicode-enabled file space.

The following example shows how this process works when automatic renaming is enabled from the server, for an existing client node that has file spaces in server storage.

1. The administrator updates a client node definition by issuing an UPDATE NODE command with the parameter, AUTOFSRENAME YES.
2. The client processes an incremental back up.
3. Tivoli Storage Manager processes the back up as follows:
   a. Renames the existing file space (_OLD)
   b. Creates a new Unicode-enabled file space
   c. Processes the back up in the current operation to the new Unicode-enabled file space

Attention: If you force the file space renaming for all clients at the same time, backups can contend for network and storage resources, and storage pools can run out of storage space.

Before you allow automatic renaming of file spaces for Unicode-enabled Tivoli Storage Manager clients, read the following sections.

“The Rules for Automatically Renaming File Spaces” on page 247

“Options for Automatically Renaming File Spaces”

“Planning for Unicode Versions of Existing Client File Spaces” on page 247

“How Clients are Affected by the Migration to Unicode” on page 249

“Example of a Migration Process” on page 250

Options for Automatically Renaming File Spaces: As an administrator, you can control whether the file spaces of any existing clients are renamed to force the creation of new Unicode-enabled file spaces. By default, no automatic renaming occurs. To control the automatic renaming, use the parameter AUTOFSRENAME when you register or update a node. You can also allow clients to make the choice. Clients can use the client option AUTOFSRENAME.

Note: The setting for AUTOFSRENAME affects only clients that are Unicode enabled.

You have these options:

- Do not allow existing file spaces to be renamed, so that Unicode-enabled file spaces are not created (AUTOFSRENAME=NO, the default).

  Tivoli Storage Manager does not automatically rename client file spaces when the client system upgrades to the Unicode-enabled Tivoli Storage Manager client. This setting can help an administrator control how many clients' file spaces can be renamed at one time. The administrator can determine how many Unicode-enabled clients exist by using the QUERY NODE FORMAT=DETAILED command. The output displays the client level. A Unicode-enabled client is on a Windows NT, Windows 2000, Windows 2002, Windows .NET, and Windows XP system at Tivoli Storage Manager Version 4.2.0 or higher.
- Automatically rename existing file spaces, forcing the creation of Unicode-enabled file spaces in place of the renamed file spaces (AUTOFSRENAME=YES).

  Tivoli Storage Manager automatically renames client file spaces in server storage when the client upgrades to the Unicode-enabled client and runs one of the following operations: archive, selective backup, full incremental backup, or partial incremental backup. Tivoli Storage Manager automatically renames the file spaces that are specified in the current operation and creates new, Unicode-enabled file spaces where files and directories are stored to complete the operation. Other file spaces that are not specified in the current operation are not affected by the rename. This means a client can have mixed file spaces. See "The Rules for Automatically Renaming File Spaces" on page 247 for how the new name is constructed.

  Attention: If you force the file space renaming for all clients at the same time, client operations can contend for network and storage resources, and storage pools can run out of storage space.
- Allow clients to choose whether to rename files spaces, in effect choosing whether new Unicode-enabled file spaces are created (AUTOFSRENAME=CLIENT).

  If you use this value for a client node, the client can set its AUTOFSRENAME option in its options file. The client option determines whether file spaces are renamed (YES or NO), or whether the user is prompted for renaming at the time of a Tivoli Storage Manager operation (PROMPT).

  The default value for the client option is PROMPT. When the option is set for prompting, the client is presented with a choice about renaming file spaces. When a client that has existing file spaces on server storage upgrades to the Unicode-enabled client, and the client runs a Tivoli Storage Manager operation with the server, the user is asked to choose whether to rename the file spaces that are involved in the current operation.

  The client is prompted only once about renaming a particular file space.

  If the client does not choose to rename the file space, the administrator can later rename the file space so that a new Unicode-enabled file space is created the next time the client processes an archive, selective backup, full incremental backup, or partial incremental backup.

  Attention: There is no prompt for operations that run with the client scheduler. If the client is running the scheduler and the client AUTOFSRENAME option is set to PROMPT, there is no prompt and the file space is not renamed. This allows a client session to run unattended. The prompt appears during the next interactive session on the client.

The following table summarizes what occurs with different parameter and option settings.

Table 21. Effects of AUTOFSRENAME Settings

Parameter on the server (for each client)	Option on the client	Result for file spaces	Is the file space renamed?
Yes	Yes, No, Prompt	Renamed	Yes
No	Yes, No, Prompt	Not renamed	No
Client	Yes	Renamed	Yes
	No	Not renamed	Yes
	Prompt	Command-line or GUI: The user receives a one-time only prompt about renaming	Depends on the response from the user (yes or no)
		Client Scheduler: Not renamed (prompt appears during the next command-line or GUI session)	No

The Rules for Automatically Renaming File Spaces: With its automatic renaming function, Tivoli Storage Manager renames a file space by adding the suffix _OLD. For example:

Original file space	\\maria\c$
Renamed file space	\\maria\c$_OLD

If the new name would conflict with the name of another file space, a number is added to the suffix. For example:

Original file space	\\maria\c$	Other existing file spaces: \\maria\c$_OLD \\maria\c$_OLD1
Renamed file space	\\maria\c$_OLD2	

If the new name for the file space exceeds the limit of 64 characters, the file space name is truncated on the right before the suffix _OLD is added.

Planning for Unicode Versions of Existing Client File Spaces: You need to consider the following factors in your planning:

- After clients with existing file spaces start to create Unicode-enabled file spaces, they will still need to have access to the renamed file spaces that are not Unicode-enabled for some period of time.
- Your storage pool and database space requirements can double if you allow all clients to create Unicode-enabled file spaces in addition to their existing file spaces that are not Unicode-enabled.
- Because the initial backups after migration are complete backups, it can also greatly increase the time required to finish backup operations.

To minimize problems, you need to plan the storage of Unicode-enabled file spaces for clients that already have existing file spaces in server storage.

1. Determine which clients need to migrate.

   Clients that have had problems with backing up files because their file spaces contain names of directories or files that cannot be converted to the server's code page should have the highest priority. Balance that with clients that are most critical to your operations. If you have a large number of clients that need to become Unicode enabled, you can control the migration of the clients.

Change the rename option for a few clients at a time to keep control of storage space usage and processing time. Also consider staging migration for clients that have a large amount of data backed up.

2. Allow for increased backup time and network resource usage when the Unicode-enabled file spaces are first created in server storage.

   Based on the number of clients and the amount of data those clients have, consider whether you need to stage the migration. Staging the migration means setting the AUTOFSRENAME parameter to YES or CLIENT for only a small number of clients every day.

   Note: If you set the AUTOFSRENAME parameter to CLIENT, be sure to have the clients (that run the client scheduler) set their option to AUTOFSRENAME YES. This ensures the file spaces are renamed.
3. Check the current storage usage for the clients that need to become Unicode enabled.

   You can use the QUERY OCCUPANCY command to display information on how much space each client is currently using. Initially, clients will need only the amount of space used by active files. Therefore, you need to estimate how much of the current space is used by copies (different versions of the same file). Migration will result in a complete backup at the next incremental backup, so clients will need space for that backup, plus for any other extra versions that they will keep. Therefore, the amount of storage required also depends on policy (see the next step). Your Tivoli Storage Manager policy specifies how files are backed up, archived, migrated from client node storage, and managed in server storage.
4. Understand how your Tivoli Storage Manager policies affect the storage that will be needed.

   If your policies expire files based only on the number of versions (Versions Data Exists), storage space required for each client will eventually double, until you delete the old file spaces.

   If your policies expire files based only on age (Retain Extra Versions), storage space required for each client will increase initially, but will not double.

   If your policies use both the number of versions and their age, each client will need less than double their current usage.
5. Estimate the effect on the database size.

   The database size depends on the number of files in server storage, as well as the number of versions of those files. As Unicode-enabled file spaces are backed up, the original file spaces that were renamed remain. Therefore, the server requires additional space in the database to store information about the increased number of file spaces and files.

   See “Estimating and Monitoring Database and Recovery Log Space Requirements” on page 386.
6. Arrange for the additional storage pool space, including space in copy storage pools, based on your estimate from step 3 and 4.
7. Check the server database space that is available and compare with your estimate from step 5.
8. Ensure that you have a full database backup *before* you proceed with migration of Unicode-enabled file spaces. See “Backing Up the Database” on page 510.

9. Consider how you will manage the renamed file spaces as they age. The administrator can delete them, or the clients can be allowed to delete their own file spaces.

How Clients are Affected by the Migration to Unicode: The server manages a Unicode-enabled client and its file spaces as follows:

- When a client upgrades to a Unicode-enabled client and logs in to the server, the server identifies the client as Unicode-enabled.

  Note: That same client (same node name) cannot log in to the server with a previous version of Tivoli Storage Manager or a client that is not Unicode-enabled.
- The original file space that was renamed (_OLD) remains with both its active and inactive file versions that the client can restore if needed. The original file space will no longer be updated. The server will not mark existing active files inactive when the same files are backed up in the corresponding Unicode-enabled file space.

  Note: Before the Unicode-enabled client is installed, the client can back up files in a code page other than the current locale, but cannot restore those files. After the Unicode-enabled client is installed, if the same client continues to use file spaces that are not Unicode-enabled, the client skips files that are not in the same code page as the current locale during a backup. Because the files are skipped, they appear to have been deleted from the client. Active versions of the files in server storage are made inactive on the server. When a client in this situation is updated to a Unicode-enabled client, you should migrate the file spaces for that client to Unicode-enabled file spaces.
- The server does not allow a Unicode-enabled file space to be sent to a client that is not Unicode enabled during a restore or retrieve process.
- Clients should be aware that they will not see all their data on the Unicode-enabled file space until a full incremental backup has been processed.

  When a client performs a selective backup of a file or directory and the original file space is renamed, the new Unicode-enabled file space will contain only the file or directory specified for that backup operation. All other directories and files are backed up on the next full incremental backup.

  If a client needs to restore a file *before* the next full incremental backup, the client can perform a restore from the renamed file space instead of the new Unicode-enabled file space. For example:

  1. Sue had been backing up her file space, \\sue-node\d$.
  2. Sue upgrades the Tivoli Storage Manager client on her system to the Unicode-enabled Tivoli Storage Manager client.
  3. Sue performs a selective backup of the file HILITE.TXT.
  4. The automatic file space renaming function is in effect and Tivoli Storage Manager renames`\\sue-node\d$` to `\\sue-node\d$_OLD`. Tivoli Storage Manager then creates a new Unicode-enabled file space on the server with the name \\sue-node\d$. This new Unicode-enabled file space contains only the HILITE.TXT file.
  5. All other directories and files in Sue's file system will be backed up on the next full incremental backup. If Sue needs to restore a file before the next full incremental backup, she can restore the file from the \\sue-node\d$_OLD file space.

Refer to the *Using the Backup-Archive Client* publication for more information.

Example of a Migration Process: This section gives one possible sequence for migrating clients. Assumptions for this scenario are:

- The Tivoli Storage Manager server database has been backed up.
- The latest server software has been installed. This installation has also performed an upgrade to the server database.
- Clients have installed the latest software.
- A few clients are file servers. Most clients are workstations used by individuals.
- Clients generally run scheduled incremental backups every night.

The following is a possible migration process:

1. Have all clients install the Unicode-enabled Tivoli Storage Manager client software.
2. Migrate the file servers first. For clients that are file servers, update the AUTOFSRENAME parameter to enable automatic renaming for the file spaces. For example, if the client node names for all file servers begin with FILE, enter the following command:

   ```
   update node file* autofsrename=yes
   ```

   This forces the file spaces to be renamed at the time of the next backup or archive operation on the file servers. If the file servers are large, consider changing the renaming parameter for one file server each day.
3. Allow backup and archive schedules to run as usual. Monitor the results.
   a. Check for the renamed file spaces for the file server clients. Renamed file spaces have the suffix `_OLD` or `_OLDn`, where *n* is a number. (See “The Rules for Automatically Renaming File Spaces” on page 247.)
   b. Check the capacity of the storage pools. Add tape or disk volumes to storage pools as needed.
   c. Check database usage statistics to ensure you have enough space.
4. Migrate the workstation clients. For example, migrate all clients with names that start with the letter *a*.

   ```
   update node a* autofsrename=yes
   ```
5. Allow backup and archive schedules to run as usual that night. Monitor the results.
6. After sufficient time passes, consider deleting the old, renamed file spaces. See “Managing the Renamed File Spaces”.

Managing the Renamed File Spaces: The file spaces that were automatically renamed (_OLD) to allow the creation of Unicode-enabled file spaces continue to exist on the server. Users can still access the file versions in these file spaces.

Because a renamed file space is not backed up again with its new name, the files that are active (the most recent backup version) in the renamed file space remain active and never expire. The inactive files in the file space expire according to the policy settings for how long versions are retained. To determine how long the files are retained, check the values for the parameters, Retain Extra Versions and Retain Only Versions, in the backup copy group of the management class to which the files are bound.

When users no longer have a need for their old, renamed file spaces, you can delete them. If possible, wait for the longest retention time for the only version

(Retain Only Version) that any management class allows. If your system has storage constraints, you may need to delete these file spaces before that.

Querying Unicode-enabled File Spaces

You can determine which file spaces are Unicode-enabled by querying all of the file spaces:

```
query filespace
```

```
Node Name  Filespace   FSID  Platform  Filespace    Is     Capacity   Pct
           Name                        Type      Filespace     (MB)  Util
                                                 Unicode?
---------- ----------- ----  -------  --------- --------- -------- -----
SUE        \\sue\c$       1  WinNT     NTFS        Yes     2,502.3  75.2
SUE        \\sue\d$       2  WinNT     NTFS        Yes     6,173.4  59.6
JOE        \\joe\c$       1  WinNT     NTFS        No     12,299.7  31.7
```

To query a specific Unicode-enabled file space, it may be more convenient to use the file space identifier (FSID) than the file space name. File space names for Unicode-enabled file spaces may not be readable when displayed in the server's code page. Attempting to enter the name of a Unicode-enabled file space may not work because it depends on the server's code page and conversion routines that attempt to convert from the server's code page to Unicode. See "Displaying Information about File Spaces" for details.

Unicode-enabled Clients and Existing Backup Sets

A client can have a backup set that contains both file spaces that are Unicode-enabled and file spaces that are not Unicode-enabled. The client must have the same level of Tivoli Storage Manager or higher to restore the data in the backup set. For example, a Version 4.1.0 client backs up file spaces, and then upgrades to Version 4.2.0 with support for Unicode-enabled file spaces. That same client can still restore the non-Unicode file spaces from the backup set.

Unicode-enabled file spaces in a backup set can only be accessed by a Unicode-enabled client, and not by an earlier version of the client. The server allows only Unicode-enabled clients to restore data from Unicode-enabled file spaces. For information about restoring backup sets, see "Restoring Backup Sets from a Backup-Archive Client" on page 318.

Displaying Information about File Spaces

You can display file space information to:

- Identify file spaces defined to each client node, so that you can delete each file space from the server before removing the client node from the server
- Identify file spaces that are Unicode enabled and identify their file space ID (FSID)
- Monitor the space used on workstation's disks
- Monitor whether backups are completing successfully for the file space
- Determine the date and time of the last backup

You display file space information by identifying the client node name and file space name.

Note: File space names are case-sensitive and must be entered exactly as known to the server.

For example, to view information about file spaces defined for client node JOE, enter:

```
query filespace joe *
```

The following figure shows the output from this command.

```
Node Name  Filespace   FSID  Platform  Filespace    Is     Capacity   Pct
           Name                        Type      Filespace     (MB)  Util
                                                 Unicode?
---------- ----------- ----  -------  --------- --------- -------- -----
JOE        \\joe\c$       1  WinNT     NTFS        Yes     2,502.3  75.2
JOE        \\joe\d$       2  WinNT     NTFS        Yes     6,173.4  59.6
```

When you display file space information in detailed format, the Filespace Name field may display file space names as "...". This indicates to the administrator that a file space does exist but could not be converted to the server's code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the administrator's Web interface or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space name or file name may display with a combination of invalid characters or blank spaces. Refer to *Administrator's Reference* for details.

Moving Data by Node

You can move a node's data in a sequential-access storage pool or move selected file spaces for a single node. For more information see, "Moving Data by Node" on page 215.

Deleting File Spaces and Client Nodes

You can delete a client node from a server, but first you must delete all of that client's data from server storage by deleting any file spaces that belong to the node.

Deleting a File Space

Administrators may want to delete a file space when:

- Users are not authorized to delete backed up or archived files in storage pools.

  The authority to delete backed up or archived files from server storage is set when a client node is registered. See "Accepting Default Closed Registration or Enabling Open Registration" on page 228 for information on allowing users to delete files in storage pools.

  For example, client node PEASE no longer needs archived files in file space */home/pease/dir2*. However, he does not have the authority to delete those files. You can delete them by entering:

  ```
  delete filespace pease /home/pease/dir2 type=archive
  ```

- You want to remove a client node from the server

  You must delete a user's files from storage pools before you can remove a client node. For example, to delete all file spaces belonging to client node ID DEBBYG, enter:

  ```
  delete filespace debbyg * type=any
  ```

- You want to delete a specific user's files

  For client nodes that support multiple users, such as UNIX, a file owner name is associated with each file on the server. The owner name is the user ID of the operating system, such as the UNIX user ID. When you delete a file space belonging to a specific owner, only files that have the specified owner name in the file space are deleted.

When a node has more than one file space and you issue a DELETE FILESPACE command for only one file space, a QUERY FILESPACE command for the node during the delete process shows no file spaces. When the delete process ends, you can view the remaining file spaces with the QUERY FILESPACE command.

Note: After you delete all of a client node's file spaces, you can delete the node with the REMOVE NODE command. See "Deleting Client Nodes" on page 237 for more details.

Modifying Client Option Files

Client nodes connect with the server through a client options file *(dsm.opt)*. This file, located in the client directory, contains client options that control processing and connections with the server. The most important option is the network address of the server, but you can add many other client options at any time. See more information about client option files, see "Connecting Nodes with the Server" on page 231.

Administrators can also create client option sets to be used in conjunction with client option files. See "Creating Client Option Sets from the Server" for more details.

All Nodes

All client options files (*dsm.opt*) can be edited with a text editor. Anyone can edit the client options file if they have access to the directory where the node software is installed. Editing individual options files is the most direct method, but may not be suitable for sites with many client nodes.

Note: If any changes are made to the dsm.opt file, the client must be restarted for changes in the options file to have any affect.

Creating Client Option Sets from the Server

An administrator can create a set of client options to be used by a client node at Tivoli Storage Manager Version 3 or later. The client options specified in the set are used in conjunction with the client options file described in "Connecting Nodes with the Server" on page 231.

Client option sets allow the administrator to specify additional options that may not be included in the client's option file (dsm.opt). You can specify which clients use the option set with the REGISTER NODE or UPDATE NODE commands. The client can use these defined options during a backup, archive, restore, or retrieve process. See *Backup-Archive Installation and User's Guide* for detailed information about individual client options.

To create a client option set and have the clients use the option set, do the following:

1. Create the client option set with the DEFINE CLOPTSET command.

2. Add client options to the option set with the DEFINE CLIENTOPT command.
3. Specify which clients should use the option set with the REGISTER NODE or UPDATE NODE command.

Creating a Client Option Set

When you create a client option set, you define a name for the option set, and can optionally provide a description of the option set. For example:

```
define cloptset engbackup description='Backup options for eng. dept.'
```

Note: The option set is empty when it is first defined.

Adding Client Options in an Option Set

You can add client options in a defined client option set.

The following example shows how to add a client option in the ENGBACKUP option set.

```
define clientopt engbackup schedlogretention 5
```

For a list of valid client options you can specify, refer to *Administrator's Reference*.

The server automatically assigns sequence numbers to the specified options, or you can choose to specify the sequence number for order of processing. This is helpful if you have defined more than one of the same option as in the following example.

```
define clientopt engbackup inclexcl "include d:\admin"
define clientopt engbackup inclexcl "include d:\payroll"
```

A sequence number of 0 is assigned to the option *include d:\admin*. A sequence number of 1 is assigned to the option *include d:\payroll*. If you want to specifically process one option before another, include the sequence parameter as follows:

```
define clientopt engbackup inclexcl "include d:\admin" seqnumber=2"
define clientopt engbackup inclexcl "include d:\payroll" seqnumber=1"
```

The options are processed starting with the highest sequence number.

Any include-exclude statements in the server client option set have priority over the include-exclude statements in the local client options file. The server include-exclude statements are always enforced and placed at the bottom of the include-exclude list and evaluated before the client include-exclude statements. If the server option set has several include-exclude statements, the statements are processed starting with the highest sequence number. The client can use the QUERY INCLEXCL command to view the include-exclude statements in the order they are processed. QUERY INCLEXCL also displays the source of each include-exclude statement. For more information on the processing of the include-exclude statements see "The Include-Exclude List" on page 280 and also the *Backup-Archive Installation and User's Guide*.

The FORCE parameter allows an administrator to specify whether a client node can override an option value. This parameter has no effect on additive options such as INCLEXCL and DOMAIN. The default value is NO. If FORCE=YES, the client cannot override the value. The following example shows how you can prevent a client from using subfile backup:

```
define clientopt engbackup subfilebackup no force=yes
```

Registering Client Nodes and Assigning Them to an Option Set

You can register or update a client node and specify an option set for the client to use as follows:

```
register node mike pass2eng cloptset=engbackup
```

The client node MIKE is registered with the password pass2eng. When the client node MIKE performs a scheduling operation, his schedule log entries are kept for 5 days.

Managing Client Options from the Server Using Client Option Sets

Administrators can perform the following activities when managing client option sets:

Task	Required Privilege Class
Updating the sequence number for a client option	System or unrestricted policy
Deleting an option from a client option set	System, unrestricted policy, or restricted policy
Copying a client option set	System, unrestricted policy, or restricted policy
Displaying client option set information	Any administrator
Updating the client option set description	System, unrestricted policy, or restricted policy
Deleting a client option set	System, unrestricted policy, or restricted policy

Updating the Sequence Number for a Client Option

You can update the sequence number for a client option to change its processing order. This is helpful if you have more than one of the same option, for example several INCLUDE options.

The following example shows how to change the sequence number for the DATEFORMAT option from 0 to 9:

```
update clientopt engbackup dateformat 0 9
```

Deleting an Option from a Client Option Set

You can remove an option that is defined in a client option set. The following example shows how to remove the SCHEDMODE polling option from the financeschd option set:

```
delete clientopt financeschd schedmode
```

Copying a Client Option Set

You can copy an existing client option to another option set. The following example shows how to copy the engbackup option set to financeschd option set:

```
copy cloptset engbackup financeschd
```

Requesting Information about a Client Option Set

To display information about the contents of a client option set, issue the following command:

```
query cloptset financeschd
```

Updating the Description for a Client Option Set

You can update the description for a client option set. The following example shows how to update the description for the engbackup option set:

```
update clopset engbackup description='Scheduling information'
```

Deleting a Client Option Set

When you delete a client option set, client node references to the option set are null. The clients continue to use their existing client options file. The following example shows how to delete the engbackup client option set:

```
delete cloptset engbackup
```

Managing Tivoli Storage Manager Sessions

Each time an administrator or client node connects with the server, an administrative or client session is established. Tivoli Storage Manager tracks its sessions in the server database. Backup-archive clients are eligible for client restartable restore sessions, however, application clients are not. See “Managing Client Restartable Restore Sessions” on page 259 for more information.

Administrators can perform the following activities when managing Tivoli Storage Manager sessions:

Task	Required Privilege Class
Displaying information about client sessions	Any administrator
Canceling a client session	System or operator
Disabling or enabling a client session	System or operator
Freeing links for client connections	Administrator with root authority

Displaying Information about Tivoli Storage Manager Sessions

Each client session is assigned a unique session number. To display information about client sessions, enter:

```
query session
```

Figure 41 shows a sample client session report.

```
  Sess Comm.  Sess     Wait   Bytes   Bytes Sess  Platform Client Name
Number Method State    Time    Sent   Recvd Type
------ ------ ------ ------ ------- ------- ----- -------- -----------
   471 Tcp/Ip IdleW   36 S      592     186 Node  WinNT    JOEUSER
   472 Tcp/Ip RecvW    0 S      730 838.2 K Node  WinNT    STATION1
   475 HTTP   Run      0 S        0       0 Admin WebBrow- ADMIN
                                                  ser
```

Figure 41. Information about Client Sessions

You can determine the state of the server by examining the *session state* and *wait time* to determine how long (in seconds, minutes, or hours) the session has been in the current state.

Server Session States

The server session state can be one of the following:

Start Connecting with a client session.

Run Executing a client request.

End Ending a client session.

RecvW
Waiting to receive an expected message from the client while a database transaction is in progress. A session in this state is subject to the COMMTIMEOUT limit.

SendW
Waiting for acknowledgment that the client has received a message sent by the server.

MediaW
Waiting for removable media to become available.

Aggregation can cause multiple media waits within a transaction and is indicated by one client message. For more information, see "Reclaiming Space in Sequential Access Storage Pools" on page 189.

Note: If QUERY SESSION FORMAT=DETAILED is specified, the Media Access Status field displays the type of media wait state.

IdleW Waiting for communication from the client, and a database transaction is NOT in progress. A session in this state is subject to the IDLETIMEOUT limit as specified in the server options file.

If a client does not initiate communication within the specified time limit set by the IDLETIMEOUT option in the server options file, then the server cancels the client session.

For example, if the IDLETIMEOUT option is set to 30 minutes, and a user does not initiate any operations within those 30 minutes, then the server cancels the client session. The client session is automatically reconnected to the server when it starts to send data again.

Canceling a Tivoli Storage Manager Session

You can cancel a client session with the CANCEL SESSION command and the associated session number. Canceling sessions may be necessary when a user's machine is not responding or as a prerequisite to halting the server. Administrators can display a session number with the QUERY SESSION command as described in "Displaying Information about Tivoli Storage Manager Sessions" on page 256.

Users and administrators whose sessions have been canceled must reissue their last command to access the server again.

If an operation, such as a backup or an archive process, is interrupted when you cancel the session, the server rolls back the results of the current transaction. That is, any changes made by the operation that are not yet committed to the database are undone. If necessary, the cancellation process may be delayed.

If the session is in the Run state when it is canceled, the cancel process does not take place until the session enters the SendW, RecvW, or IdleW state. For details, see "Server Session States" on page 257.

If the session you cancel is currently waiting for a media mount, the mount request is automatically canceled. If a volume associated with the client session is currently being mounted by an *automated* library, the cancel may not take effect until the mount is complete.

For example, to cancel a session for client MARIE:

1. Query client sessions to determine the session number as shown Figure 41 on page 256. The example report displays MARIE's session number 6.
2. Cancel node MARIE's session by entering:
   ```
   cancel session 6
   ```

If you want to cancel all backup and archive sessions, enter:

```
cancel session all
```

When a Client Session is Automatically Canceled

Client sessions can be automatically canceled based on the settings of the following server options:

COMMTIMEOUT
Specifies how many seconds the server waits for an expected client message during a transaction that causes a database update. If the length of time exceeds this time-out, the server rolls back the transaction that was in progress and ends the client session. The amount of time it takes for a client to respond depends on the speed and processor load for the client and the network load.

IDLETIMEOUT
Specifies how many minutes the server waits for a client to initiate communication. If the client does not initiate communication with the server within the time specified, the server ends the client session. For example, the server prompts the client for a scheduled backup operation but the client node is not started. Another example can be that the client program is idle while waiting for the user to choose an action to perform (for example, backup archive, restore, or retrieve files). If a user starts the client session and does not choose an action to perform, the session will time out. The client program automatically reconnects to the server when the user chooses an action that requires server processing. A large number of idle sessions can inadvertently prevent other users from connecting to the server.

THROUGHPUTDATATHRESHOLD
Specifies a throughput threshold, in kilobytes per second, a client session must achieve to prevent being cancelled after the time threshold is reached. Throughput is computed by adding send and receive byte counts and dividing by the length of the session. The length does not include time spent waiting for media mounts and starts at the time a client sends data to the server for storage. This option is used in conjunction with the THROUGHPUTTIMETHRESHOLD server option.

THROUGHPUTTIMETHRESHOLD
Specifies the time threshold, in minutes, for a session after which it may be canceled for low throughput. The server ends a client session when it has

been active for more minutes than specified and the data transfer rate is less than the amount specified in the THROUGHPUTDATATHRESHOLD server option.

Refer to the *Administrator's Reference* for more information.

Disabling or Enabling Access to the Server

Task	Required Privilege Class
Disabling and enabling client node access to the server	System or operator
Displaying server status	Any administrator

You can prevent clients from establishing sessions with the server by using the DISABLE SESSIONS command. This command does not cancel sessions currently in progress or system processes such as migration and reclamation. For example, to disable client node access to the server, enter:

```
disable sessions
```

You continue to access the server and current client activities complete unless a user logs off or an administrator cancels a client session. After the client sessions have been disabled, you can enable client sessions and resume normal operations by entering:

```
enable sessions
```

You can issue the QUERY STATUS command to determine if the server is enabled or disabled.

Managing Client Restartable Restore Sessions

Some large restore operations may invoke a special type of restore operation called client restartable restore sessions. These special sessions allow users to restart the restore session from where it left off if the session was interrupted. Tivoli Storage Manager identifies client restartable restore sessions by displaying message ANS1247I on the client machine when the sessions start. These restore sessions can be restarted as long as the restore interval has not expired.

When a restartable restore session is saved in the server database the file space is locked in server storage. The following is in effect during the file space lock:

- Files residing on sequential volumes associated with the file space cannot be moved.
- Files associated with the restore cannot be backed up. However, files not associated with the restartable restore session that are in the same file space are eligible for backup. For example, if you are restoring all files in directory A, you can still backup files in directory B from the same file space.

The RESTOREINTERVAL server option allows administrators to specify how long client restartable restore sessions are saved in the server database. Consider scheduled backup operations when setting this option. For more information, refer to the RESTOREINTERVAL server option in *Administrator's Reference*..

Administrators can perform the following activities when managing client restartable restore sessions:

Task	Required Privilege Class
Displaying information about client restartable restore sessions	Any administrator
Canceling client restartable restore sessions	System or operator
Interrupting client restartable restore sessions	System or operator

Displaying Information about a Client Restartable Restore Session

You can display information about client restartable restore sessions with the QUERY RESTORE command. For example, to determine which client nodes have eligible restartable restore sessions, enter:

```
query restore
```

Restartable restore sessions have a negative session number.

Canceling a Client Restartable Restore Session

When a client restore session is in a restartable state, the file space is locked in server storage and no files can be moved from sequential volumes. This prevents the data from being migrated, moved, reclaimed, or backed up by another operation. These sessions will automatically expire when the specified restore interval has passed.

An administrator can cancel a restartable restore session that is in an active or restartable state. If the restore session is active, any outstanding mount requests related to the active session are automatically canceled. When a restartable restore session is canceled with the CANCEL RESTORE command, it cannot be restarted from the point of interruption. A restartable restore session always has a negative session number.

To cancel a restartable restore session, you must specify the session number. For example:

```
cancel restore -1
```

Interrupting an Active Client Restartable Restore Session

An administrator can interrupt an active restartable restore session and have the option to later restart the session from its point of interruption by canceling the session.

```
cancel session -2
```

Managing Tivoli Storage Manager Security

Administrators can perform the following activities when managing Tivoli Storage Manager security.

Task
Managing administrators
Managing levels of administrative authority
Managing administrator access to the server and clients
Managing passwords
Managing the server console

Managing Tivoli Storage Manager Administrators

The administrator is responsible for registering other administrators, granting levels of authority, renaming or removing administrators, or for locking and unlocking administrators from the server.

Task	Required Privilege Class
Registering an administrator	System
Granting administrative authority	System
Updating information about other administrators	System
Updating information about yourself	Any administrator
Displaying information about administrators	Any administrator
Renaming an administrator user ID	System
Removing administrators	System
Locking or unlocking administrators from the server	System

About the Server Console

At installation, the server console is defined with a special user ID, which is named SERVER_CONSOLE. This name is reserved and cannot be used by another administrator.

An administrator with system privilege can revoke or grant new privileges to the SERVER_CONSOLE user ID. However, an administrator cannot update, lock, rename, or remove the SERVER_CONSOLE user ID. The SERVER_CONSOLE user ID does not have a password. Therefore, you cannot use the user ID from an administrative client unless you set authentication off.

Registering Administrators

The administrator registers other administrators with the REGISTER ADMIN command.

To register the administrator with a user ID of DAVEHIL and the password of *birds*, and a password expiration period of 120 days, enter the REGISTER ADMIN command:

```
register admin davehil birds passexp=120 contact='backup team'
```

Granting an Administrator Privilege Class Authority

After administrators are registered, they can make queries and request command-line help. To perform other server functions, they must be granted authority by being assigned one or more administrative privilege classes.

This section describes the privilege classes, which are illustrated in Figure 42 on page 262. An administrator with system privilege can perform any server function. Administrators with policy, storage, operator, analyst, or node privileges can perform subsets of server functions.

When an administrator accesses the administrative Web interface, only the tasks that correspond to the administrator's privilege class are displayed.

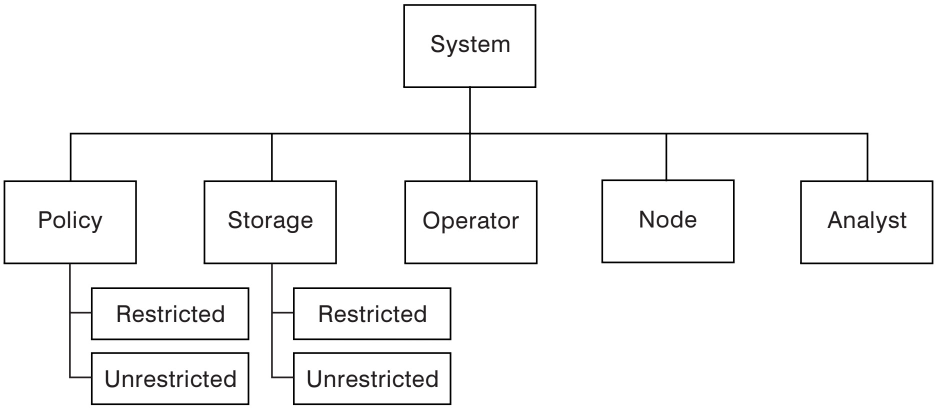


Figure 42. Administrative Privilege Classes

Privilege classes, and examples of how to set privilege classes, can be summarized as follows:

Privilege Class	Responsibilities
System `grant authority rocko classes=system`	Can perform any server administrative task. • System-wide responsibilities • Manage the enterprise • Manage Tivoli Storage Manager security
Unrestricted Policy `grant authority smith classes=policy`	Can manage the backup and archive services for nodes assigned to any policy domain. • Manage nodes • Manage policy • Manage schedules
Restricted Policy `grant authority jones domains=engpoldom`	Same responsibilities as unrestricted policy except authority is limited to specific policy domains.
Unrestricted Storage `grant authority coyote classes=storage`	Can manage server storage, but cannot define or delete storage pools. • Manage the database and recovery log • Manage Tivoli Storage Manager devices • Manage Tivoli Storage Manager storage
Restricted Storage `grant authority holland stgpools=tape*`	Can manage server storage but is limited to specific storage pools. • Manage Tivoli Storage Manager devices • Manage Tivoli Storage Manager storage
Operator `grant authority bill classes=operator`	Can control the immediate operation of the server and the availability of storage media. • Manage the Tivoli Storage Manager server • Manage client sessions • Manage tape operations
Node `grant authority help1 classes=node node=labclient`	Can access a Web backup-archive client to perform backup and restore operations.

Privilege Class	Responsibilities
Analyst `grant authority marysmith classes=analyst`	Can reset the counters that track Tivoli Storage Manager server statistics.

Updating Information about Other Administrators

An administrator can reset another administrator's password with the UPDATE ADMINISTRATOR command. For example, administrator DAVEHIL changes his password to *ganymede*, by issuing the following command:

```
update admin davehil ganymede
```

Renaming an Administrator

You can rename an administrator ID when an employee wants to be identified by a new ID, or you want to assign an existing administrator ID to another person. You cannot rename an administrator ID to one that already exists on the system.

For example, if administrator HOLLAND leaves your organization, you can assign administrative privilege classes to another user by completing the following steps:

1. Assign HOLLAND's user ID to WAYNESMITH by issuing the RENAME ADMIN command:

   ```
   rename admin holland waynesmith
   ```

   By renaming the administrator's ID, you remove HOLLAND as a registered administrator from the server. In addition, you register WAYNESMITH as an administrator with the password, contact information, and administrative privilege classes previously assigned to HOLLAND.
2. Change the password to prevent the previous administrator from accessing the server by entering:

   ```
   update admin waynesmith new_password contact="development"
   ```

Note: The administrator SERVER_CONSOLE cannot be renamed. See "About the Server Console" on page 261.

Removing Administrators

You can remove administrators from the server so that they no longer have access to administrator functions. For example, to remove registered administrator ID SMITH, enter:

```
remove admin smith
```

Notes:
1. You cannot remove the last system administrator from the system.
2. You cannot remove the administrator SERVER_CONSOLE. See "About the Server Console" on page 261 for more information.

Displaying Information about Administrators

Any administrator can query the server to display administrator information. You can also query all administrators authorized with a specific privilege class.

For example, to query the system for a detailed report on administrator ID DAVEHIL, issue the QUERY ADMIN command:

```
query admin davehil format=detailed
```

Figure 43 on page 264 displays a detailed report.

```
        Administrator Name: DAVEHIL
     Last Access Date/Time: 1998.06.04 17.10.52
    Days Since Last Access: <1
    Password Set Date/Time: 1998.06.04 17.10.52
   Days Since Password Set: 26
     Invalid Sign-on Count: 0
                   Locked?: No
                   Contact:
          System Privilege: Yes
          Policy Privilege: **Included with system privilege**
         Storage Privilege: **Included with system privilege**
         Analyst Privilege: **Included with system privilege**
        Operator Privilege: **Included with system privilege**
   Client Access Privilege: **Included with system privilege**
    Client Owner Privilege: **Included with system privilege**
    Registration Date/Time: 05/09/1998 23:54:20
 Registering Administrator: SERVER_CONSOLE
          Managing profile:
Password Expiration Period:  90 Day (s)
```

Figure 43. A Detailed Administrator Report

Locking and Unlocking Administrators from the Server

Administrators can prevent other administrators from accessing the server by locking and unlocking their administrative privilege classes. For details, see "Locking and Unlocking Administrators from the Server" on page 265.

Managing Levels of Administrative Authority

A privilege class is a level of authority granted to an administrator. The privilege class determines which administrative tasks the administrator can perform. See "Granting an Administrator Privilege Class Authority" on page 261 and *Administrator's Reference* about the activities that administrators can perform with each privilege class.

You can perform the following activities when managing other administrators' levels of authority:

Task	Required Privilege Class
Modifying administrators level of authority	System
Locking and unlocking administrators from the server	System

Modifying Administrator Levels of Authority

You may need to modify other administrators levels of authority as more clients and administrators are added to the Tivoli Storage Manager environment. If a person already has some level of authority, granting additional authority adds to any existing privilege classes; it does not override those classes.

Extending Authority for Administrators: You can grant and extend authority with the GRANT AUTHORITY command. For example, JONES has restricted policy privilege for policy domain ENGPOLDOM. Enter the following command to extend JONES' authority to policy domain MKTPOLDOM and add operator privilege:

```
grant authority jones domains=mktpoldom classes=operator
```

As an additional example, assume that three tape storage pools exist: TAPEPOOL1, TAPEPOOL2, and TAPEPOOL3. To grant restricted storage privilege for these storage pools to administrator HOLLAND, you can enter the previous command:

```
grant authority holland stgpools=tape*
```

HOLLAND is restricted to managing storage pools beginning with TAPE that existed when the authority was granted. HOLLAND is not authorized to manage any storage pools that are defined after authority has been granted.

To add a new storage pool, TAPEPOOL4, to HOLLAND's authority, enter:

```
grant authority holland stgpools=tapepool4
```

Reducing Authority for Administrators: You can revoke part of an administrator's authority with the REVOKE AUTHORITY command and specifying the administrator's ID and one or more privilege classes.

Assume that rather than revoking all of the privilege classes for administrator JONES you wished only to revoke his operator authority and his policy authorization to policy domain MKTPOLDOM. You would enter:

```
revoke authority jones classes=operator domains=mktpoldom
```

JONES still has policy privilege to the ENGPOLDOM policy domain.

Reducing Privilege Classes: You can reduce an administrator's authority simply by revoking one or more privilege classes and granting one or more other classes.

For example, administrator HOGAN has system authority. To reduce HOGAN to the operator privilege class do the following:

1. Revoke the system privilege class by entering:

   ```
   revoke authority hogan classes=system
   ```

2. Grant operator privilege class by entering:

   ```
   grant authority hogan classes=operator
   ```

Revoking Authority for Administrators: You can revoke an administrator's authority with the REVOKE AUTHORITY command. To revoke all administrative privilege classes, do not specify any privilege classes, policy domains, or storage pools. For example, to revoke both the storage and operator privilege classes from administrator JONES enter:

```
revoke authority jones
```

Locking and Unlocking Administrators from the Server: You can lock out other administrators to temporarily prevent them from accessing Tivoli Storage Manager with the LOCK ADMIN command.

For example, administrator MARYSMITH takes a leave of absence from your business. You can lock her out by entering:

```
lock admin marysmith
```

When she returns, any system administrator can unlock her administrator ID by entering:

```
unlock admin marysmith
```

MARYSMITH can now access the server to complete administrative tasks.

You cannot lock or unlock the SERVER_CONSOLE ID from the server. See "About the Server Console" on page 261 for details.

Managing Access to the Server and Clients

An administrator can control access to the server by registering and granting authority to administrators, renaming or removing an administrator, or by locking and unlocking an administrator from the server.

By default, a system or policy administrator over a specified client's domain can create a backup set from a client node's latest active files. For more information, see "Chapter 15. Managing Schedules for Client Nodes" on page 333.

Preventing Clients from Accessing the Server

You can prevent clients from establishing administrative sessions with the server. For details, see "Locking and Unlocking Client Nodes" on page 236.

Preventing Administrators from Accessing the Server

You can prevent other administrators from establishing administrative sessions with the server. For details, see "Locking and Unlocking Administrators from the Server" on page 265.

Disabling or Enabling Client Sessions

You can prevent clients from establishing sessions with the server. This effectively locks the nodes from the server. For details, see "Disabling or Enabling Access to the Server" on page 259.

Managing Passwords

By default, Tivoli Storage Manager requires authorized administrators and nodes to identify themselves to the server with a password.

Administrators can perform the following activities when managing passwords

Task	Required Privilege Class
Modifying the default timeout period for the administrative Web interface Modifying the default password expiration period Setting the limit for invalid password attempts Setting the minimum limit for passwords Disabling the default password authentication	System

Modifying the Default Timeout Period for the Administrative Web Interface

At installation, the timeout default value for the administrative Web interface is 10 minutes. When the timeout period expires, the user of the Web interface is required to reauthenticate by logging on and specifying a password. The following example shows how to set the timeout value to 20 minutes:

```
set webauthtimeout 20
```

You can specify a value from 0 to 9999 minutes. If the minimum value is 0, there is no timeout period for the administrative Web interface. To help ensure the security of an unattended browser, it is recommended that you set the timeout value higher than zero.

Modifying the Default Password Expiration Period

By default, the server sets a password expiration of 90 days. The expiration period begins when an administrator or client node is first registered to the server. If a user password is not changed within this period, the server prompts the user to change the password the next time the user tries to access the server.

To set the password expiration period for selected administrators or client nodes, you must specify the administrator or node names with the ADMIN or NODE parameter with the SET PASSEXP command. If you set the expiration period only for selected users, you may set the expiration period from 0–9999 days. A value of 0 means that user’s password never expires. For example, to set the expiration period of client node LARRY to 120 days, issue the following command:

```
set passexp 120 node=larry
```

Note: Once you have explicitly set a password expiration for a node or administrator, it is not modified if you later set a password expiration for all users.

Setting a Limit for Invalid Password Attempts

By default, Tivoli Storage Manager does not check the number of times a user attempts to log in with an invalid password. You can set a limit on consecutive invalid password attempts for all client nodes. When the limit is exceeded, the server locks the node. The following example sets a system-wide limit of three consecutive invalid password attempts:

```
set invalidpwlimit 3
```

The default value at installation is 0. A value of 0 means that invalid password attempts are not checked. You can set the value from 0 to 9999 attempts.

If you initially set a limit of 4 and then change the limit to a lower number, some clients may fail verification during the next login attempt.

After a client node has been locked, only a storage administrator with proper authority can unlock the node. For information about unlocking a client or administrator node, see “Locking and Unlocking Client Nodes” on page 236 and “Locking and Unlocking Administrators from the Server” on page 265.

An administrator can also force a client to change their password on the next login by specifying the FORCEPWRESET=YES parameter on the UPDATE NODE or UPDATE ADMIN command. For more information, refer to *Administrator’s Reference.*

Setting a Minimum Length for a Password

By default, Tivoli Storage Manager does not check the minimum length of a password. The administrator can specify a minimum password length that is required for Tivoli Storage Manager passwords. The following example shows how to set the minimum password length to eight characters:

```
set minpwlength 8
```

The default value at installation is 0. A value of 0 means that password length is not checked. You can set the length value from 0 to 64.

Disabling the Default Password Authentication

By default, the server automatically sets password authentication on. With password authentication set to on, all users must enter a password when accessing the server. To allow administrators and client nodes to access the server without entering a password, issue the following command:

```
set authentication off
```

Attention: Setting password authentication off reduces data security.

Chapter 12. Implementing Policies for Client Data

Policies are rules that you set at the Tivoli Storage Manager server to help you manage client data. Policies control how and when client data is stored, for example:

- How and when files are backed up and archived to server storage
- How space-managed files are migrated to server storage
- The number of copies of a file and the length of time copies are kept in server storage

Tivoli Storage Manager provides a standard policy that sets rules to provide a basic amount of protection for data on workstations. If this standard policy meets your needs, you can begin using Tivoli Storage Manager immediately. See “Basic Policy Planning” on page 270 for information about the standard policy.

The server process of expiration is one way that the server enforces policies that you define. Expiration processing determines when files are no longer needed, that is, when the files are expired. For example, if you have a policy that requires only four copies of a file be kept, the fifth and oldest copy is expired. During expiration processing, the server removes entries for expired files from the database, effectively deleting the files from server storage. See “File Expiration and Expiration Processing” on page 273 and “Running Expiration Processing to Delete Expired Files” on page 302 for details.

You may need more flexibility in your policies than the standard policy provides. To accommodate individual user’s needs, you may fine tune the STANDARD policy (see “Getting Users Started” on page 272 for details), or create your own policies (see “Creating Your Own Policies” on page 288 for details). Some types of clients or situations require special policy. For example, you may want to enable clients to restore backed-up files to a specific point in time (see “Setting Policy to Enable Point-in-Time Restore for Clients” on page 308 for more information).

Policy can be distributed from a configuration manager to managed servers. See “Chapter 20. Working with a Network of Tivoli Storage Manager Servers” on page 431 for more information on distributing configurations.

See the following sections:

Concepts:
“Basic Policy Planning” on page 270
“The Standard Policy” on page 271
“File Expiration and Expiration Processing” on page 273
“Client Operations Controlled by Policy” on page 273
“The Parts of a Policy” on page 275
“More on Management Classes” on page 279
“How Tivoli Storage Manager Selects Files for Policy Operations” on page 284
“How Client Migration Works with Backup and Archive” on page 288

© Copyright IBM Corp. 1993, 2002

Tasks:
"Getting Users Started" on page 272
"Changing Policy" on page 272
"Creating Your Own Policies" on page 288
"Defining and Updating a Policy Domain" on page 290
"Defining and Updating a Policy Set" on page 291
"Defining and Updating a Management Class" on page 292
"Defining and Updating a Backup Copy Group" on page 293
"Defining and Updating an Archive Copy Group" on page 298
"Assigning a Default Management Class" on page 300
"Validating and Activating a Policy Set" on page 300
"Assigning Client Nodes to a Policy Domain" on page 302
"Running Expiration Processing to Delete Expired Files" on page 302
"Setting Policy to Enable Point-in-Time Restore for Clients" on page 308
"Policy for Tivoli Storage Manager Servers as Clients" on page 308
"Configuring Policy for Tivoli Data Protection Application Clients" on page 304
"Policy for Logical Volume Backups" on page 304
"Configuring Policy for Managed System for SAN" on page 306
"Distributing Policy Using Enterprise Configuration" on page 309
"Querying Policy" on page 309
"Deleting Policy" on page 312

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Basic Policy Planning

Start out simply to plan your policy. You may be able to use the default policy that comes with the server. Ask the questions:

- How many backup versions do clients need?
- How long do clients need the backup versions?

Examine the default policy to see if it meets your needs:

- Up to two backup versions of a file on the client's system are retained in server storage.
- The most recent backup version is retained for as long as the original file is on the client file system. All other versions are retained for up to 30 days after they become inactive.
- One backup version of a file that has been deleted from the client's system is retained in server storage for 60 days.
- An archive copy is kept for up to 365 days.

See “The Standard Policy” for more details about the standard policy.

The server manages files based on whether the files are active or inactive. The most current backup or archived copy of a file is the active version. All other versions are called inactive versions. An active version of a file becomes inactive when:

- A new backup is made
- A user deletes that file on the client node and then runs an incremental backup

Policy determines how many inactive versions of files the server keeps, and for how long. When files exceed the criteria, the files expire. Expiration processing can then remove the files from the server database. See “File Expiration and Expiration Processing” on page 273 and “Running Expiration Processing to Delete Expired Files” on page 302 for details.

The Standard Policy

The standard policy consists of a standard policy domain, policy set, management class, backup copy group, and archive copy group. Each of these parts is named STANDARD. See “The Parts of a Policy” on page 275 for details. The attributes of the default policy are as follows:

Table 22. Summary of Default Policy

Policy	Object where the policy is set
Backup Policies	
Files are backed up to the default disk storage pool, BACKUPPOOL.	STANDARD backup copy group, DESTINATION parameter
An incremental backup is performed only if the file has changed since the last backup.	STANDARD backup copy group, MODE parameter
Files cannot be backed up while they are being modified.	STANDARD backup copy group, SERIALIZATION parameter
Up to two backup versions of a file on the client’s system are retained in server storage. The most recent backup version is retained for as long as the original file is on the client file system. All other versions are retained for up to 30 days after they become inactive.	STANDARD backup copy group, the following parameters: VEREXISTS RETEXTRA RETONLY
One backup version of a file that has been deleted from the client’s system is retained in server storage for 60 days.	STANDARD backup copy group, VERDELETED parameter
When a backed up file is no longer associated with a backup copy group, it remains in server storage for 30 days (backup retention grace period).	STANDARD policy domain, BACKRETENTION parameter
Archive Policies	
Files are archived in the default disk storage pool, ARCHIVEPOOL.	STANDARD archive copy group, DESTINATION parameter
Files cannot be archived while they are being modified.	STANDARD archive copy group, SERIALIZATION parameter
An archive copy is kept for up to 365 days.	STANDARD archive copy group, RETVER parameter

Table 22. Summary of Default Policy (continued)

Policy	Object where the policy is set
When an archived file is no longer associated with an archive copy group, it remains in server storage for 365 days (archive retention grace period).	STANDARD policy domain, ARCHRETENTION parameter
General	
The default management class is STANDARD.	STANDARD policy set (ACTIVE), ASSIGN DEFMGMTCLASS command
Space Management (HSM) Policy	
Client files are not space-managed (there are no HSM clients).	STANDARD management class, SPACEMGTECHNIQUE parameter

Getting Users Started

When you register a client node, the default is to assign the node to the STANDARD policy domain. If users register their own workstations during open registration, they are also assigned to the STANDARD policy domain.

To help users take advantage of Tivoli Storage Manager, you can further tune the policy environment by doing the following:

- Define sets of client options for the different groups of users. See "Creating Client Option Sets from the Server" on page 253 for details.
- Help users with creating the include-exclude list. For example:
  - Create include-exclude lists to help inexperienced users who have simple file management needs. One way to do this is to define a basic include-exclude list as part of a client option set. This also gives the administrator some control over client usage. See "Creating Client Option Sets from the Server" on page 253 for details.
  - Provide a sample include-exclude list to users who want to specify how the server manages their files. You can show users who prefer to manage their own files how to:
    - Request information about management classes
    - Select a management class that meets backup and archive requirements
    - Use include-exclude options to select management classes for their files

  For information on the include-exclude list, see the user's guide for the appropriate client. See also "The Include-Exclude List" on page 280.
- Automate incremental backup procedures by defining schedules for each policy domain. Then associate schedules with client nodes in each policy domain. For information on schedules, see "Chapter 14. Scheduling Operations for Client Nodes" on page 325.

Changing Policy

Some types of clients and situations require policy changes. For example, if you need to direct client data to storage pools different from the default storage pools, you need to change policy. Other situations may also require policy changes. See "Configuring Policy for Specific Cases" on page 303 for details.

To change policy that you have established in a policy domain, you must replace the ACTIVE policy set. You replace the ACTIVE policy set by activating another policy set. Do the following:

1. Create or modify a policy set so that it contains the policy that you want to implement.
   - Create a new policy set either by defining a new policy set or by copying a policy set.
   - Modify an existing policy set (it cannot be the ACTIVE policy set).

   Note: You cannot directly modify the ACTIVE policy set. If you want to make a small change to the ACTIVE policy set, copy the policy to modify it and follow the steps here.
2. Make any changes that you need to make to the management classes, backup copy groups, and archive copy groups in the new policy set. For details, see "Defining and Updating a Management Class" on page 292, "Defining and Updating a Backup Copy Group" on page 293, and "Defining and Updating an Archive Copy Group" on page 298.
3. Validate the policy set. See "Validating a Policy Set" on page 300 for details.
4. Activate the policy set. The contents of your new policy set becomes the ACTIVE policy set. See "Activating a Policy Set" on page 301 for details.

File Expiration and Expiration Processing

An expired file is a file that the server no longer needs to keep, according to policy. Files expire under the following conditions:
- Users delete file spaces from client nodes
- Users expire files by using the EXPIRE command on the client (client software at Version 4.2 and later)
- A file that is a backup version exceeds the criteria in the backup copy group (how long a file is kept and how many inactive versions of a file are kept)
- An archived file exceeds the time criteria in the archive copy group (how long archived copies are kept)
- A backup set exceeds the retention time that is specified for it

Note: A base file is not eligible for expiration until all of its dependent subfiles have been expired. For details, see "Expiration Processing of Base Files and Subfiles" on page 324.

The server deletes expired files from the server database only during expiration processing. After expired files are deleted from the database, the server can reuse the space in the storage pools that was occupied by expired files. You should ensure that expiration processing runs periodically to allow the server to reuse space. See "Reclaiming Space in Sequential Access Storage Pools" on page 189 and "Running Expiration Processing to Delete Expired Files" on page 302 for more information.

Expiration processing also removes from the database any restartable restore sessions that exceed the time limit set for such sessions by the RESTOREINTERVAL server option. See "Managing Client Restartable Restore Sessions" on page 259 for information about restartable restore sessions.

Client Operations Controlled by Policy

Tivoli Storage Manager policies govern the following client operations, which are discussed in this section:
- "Backup and Restore" on page 274
- "Archive and Retrieve" on page 274

- "Client Migration and Recall"

Backup and Restore

Backup-archive clients can back up and restore files and directories. Backup-archive clients on UNIX systems can also back up and restore logical volumes. Backups allow users to preserve different versions of files as they change.

Backup

To guard against the loss of information, the backup-archive client can copy files, subdirectories, and directories to media controlled by the server. Backups can be controlled by administrator-defined policies and schedules, or users can request backups of their own data. The backup-archive client provides two types of backup:

Incremental backup
The backup of files according to policy defined in the backup copy group of the management class for the files. An incremental backup typically backs up all files that are new or that have changed since the last incremental backup.

Selective backup
Backs up only files that the user specifies. The files must also meet some of the policy requirements defined in the backup copy group.

See *Backup-Archive Installation and User's Guide* for details on backup-archive clients that can also back up logical volumes. The logical volume must meet some of the policy requirements that are defined in the backup copy group.

Restore

When a user restores a backup version of a file, the server sends a copy of the file to the client node. The backup version remains in server storage. Restoring a logical volume backup works the same way.

If more than one backup version exists, a user can restore the active backup version or any inactive backup versions.

If policy is properly set up, a user can restore backed-up files to a specific time. See "Setting Policy to Enable Point-in-Time Restore for Clients" on page 308 for details on the requirements.

Archive and Retrieve

To preserve files for later use or for records retention, a user with a backup-archive client can archive files, subdirectories, and directories on media controlled by the server. When users archive files, they can choose to have the backup-archive client erase the original files from their workstation after the client archives the files.

When a user retrieves a file, the server sends a copy of the file to the client node. The archived file remains in server storage.

Client Migration and Recall

When the Tivoli Space Manager product is on the workstation, a user can migrate files from workstation storage to server storage and recall those files as needed. Tivoli Space Manager frees space for new data and makes more efficient use of your storage resources. The installed Tivoli Space Manager product is also called the space manager client or the HSM client.

Files that are migrated and recalled with the HSM client are called *space-managed* files.

For details about using Tivoli Space Manager, see *Tivoli Space Manager for UNIX: Using the Hierarchical Storage Management Clients*.

Migration

When a file is migrated to the server, it is replaced on the client node with a small stub file of the same name as the original file. The stub file contains data needed to locate the migrated file on server storage.

Tivoli Space Manager provides selective and automatic migration. Selective migration lets users migrate files by name. The two types of automatic migration are:

Threshold
: If space usage exceeds a high threshold set at the client node, migration begins and continues until usage drops to the low threshold also set at the client node.

Demand
: If an out-of-space condition occurs for a client node, migration begins and continues until usage drops to the low threshold.

To prepare for efficient automatic migration, Tivoli Space Manager copies a percentage of user files from the client node to the Tivoli Storage Manager server. The *premigration* process occurs whenever Tivoli Space Manager completes an automatic migration. The next time free space is needed at the client node, the files that have been premigrated to the server can quickly be changed to stub files on the client. The default premigration percentage is the difference between the high and low thresholds.

Files are selected for automatic migration and premigration based on the number of days since the file was last accessed and also on other factors set at the client node.

Recall

Tivoli Space Manager provides selective and transparent recall. Selective recall lets users recall files by name. Transparent recall occurs automatically when a user accesses a migrated file.

Reconciliation

Migration and premigration can create inconsistencies between stub files on the client node and space-managed files in server storage. For example, if a user deletes a migrated file from the client node, the copy remains at the server. At regular intervals set at the client node, Tivoli Storage Manager compares client node and server storage and reconciles the two by deleting from the server any outdated files or files that do not exist at the client node.

The Parts of a Policy

Policy administrators use Tivoli Storage Manager policy to specify how files are backed up, archived, migrated from client node storage, and managed in server storage. Figure 44 on page 276 shows the parts of a policy and the relationships among the parts. You may refer to “Example: Sample Policy Objects” on page 289.

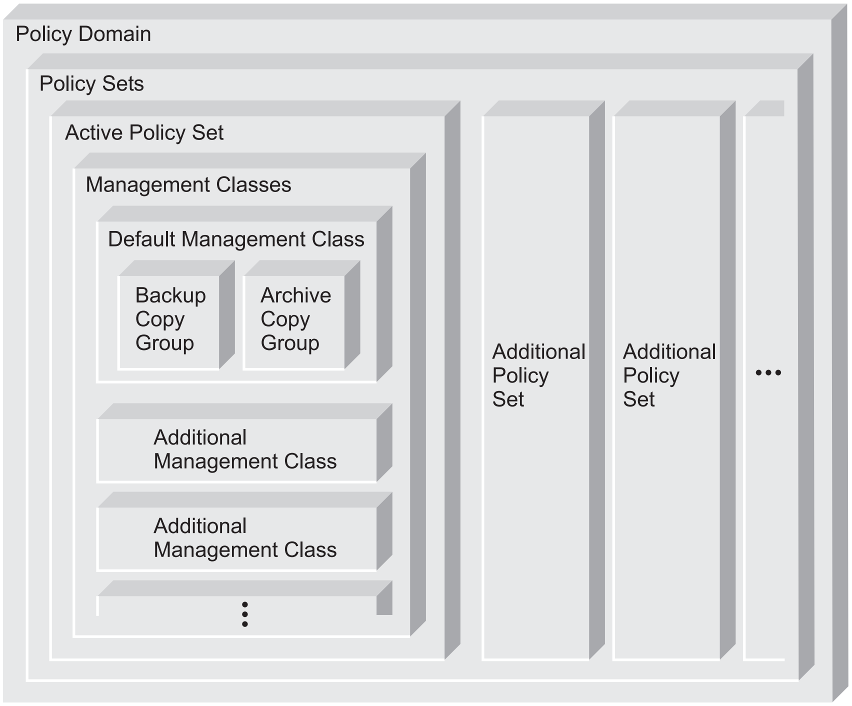


Figure 44. Tivoli Storage Manager Policy

Backup copy group
Controls the backup processing of files associated with the management class. A backup copy group determines the following:

- How frequently a file can be backed up
- How to handle files that are in use during a backup
- Where the server initially stores backup versions of files and directories
- How many backup versions the server keeps of files and directories
- How long the server keeps backup versions of files and directories, see "Running Expiration Processing to Delete Expired Files" on page 302 for details

Archive copy group
Controls the archive processing of files associated with the management class. An archive copy group determines the following:

- How to handle files that are in use during archive
- Where the server stores archived copies of files
- How long the server keeps archived copies of files, see "Running Expiration Processing to Delete Expired Files" on page 302 for details

Management class
Associates backup and archive groups with files, and specifies if and how client node files are migrated to storage pools. A management class can contain one backup or archive copy group, both a backup and archive copy group, or no copy groups. Users can *bind* (that is, associate) their files to a management class through the include-exclude list.

See "More on Management Classes" on page 279 for details.

Policy set
Specifies the management classes that are available to groups of users. Policy sets contain one or more management classes. You must identify one

management class as the *default management class*. Only one policy set, the ACTIVE policy set, controls policy operations.

Policy domain

Lets an administrator group client nodes by the policies that govern their files and by the administrators who manage their policies. A policy domain contains one or more policy sets, but only one policy set (named ACTIVE) can be active at a time. The server uses only the ACTIVE policy set to manage files for client nodes assigned to a policy domain.

You can use policy domains to:

- Group client nodes with similar file management requirements
- Provide different default policies for different groups of clients
- Direct files from different groups of clients to different storage hierarchies based on need (different file destinations with different storage characteristics)
- Restrict the number of management classes to which clients have access

Relationships among Clients, Storage, and Policy

Figure 45 on page 278 summarizes the relationships among the physical device environment, Tivoli Storage Manager storage and policy objects, and clients. The numbers in the following list correspond to the numbers in the figure.

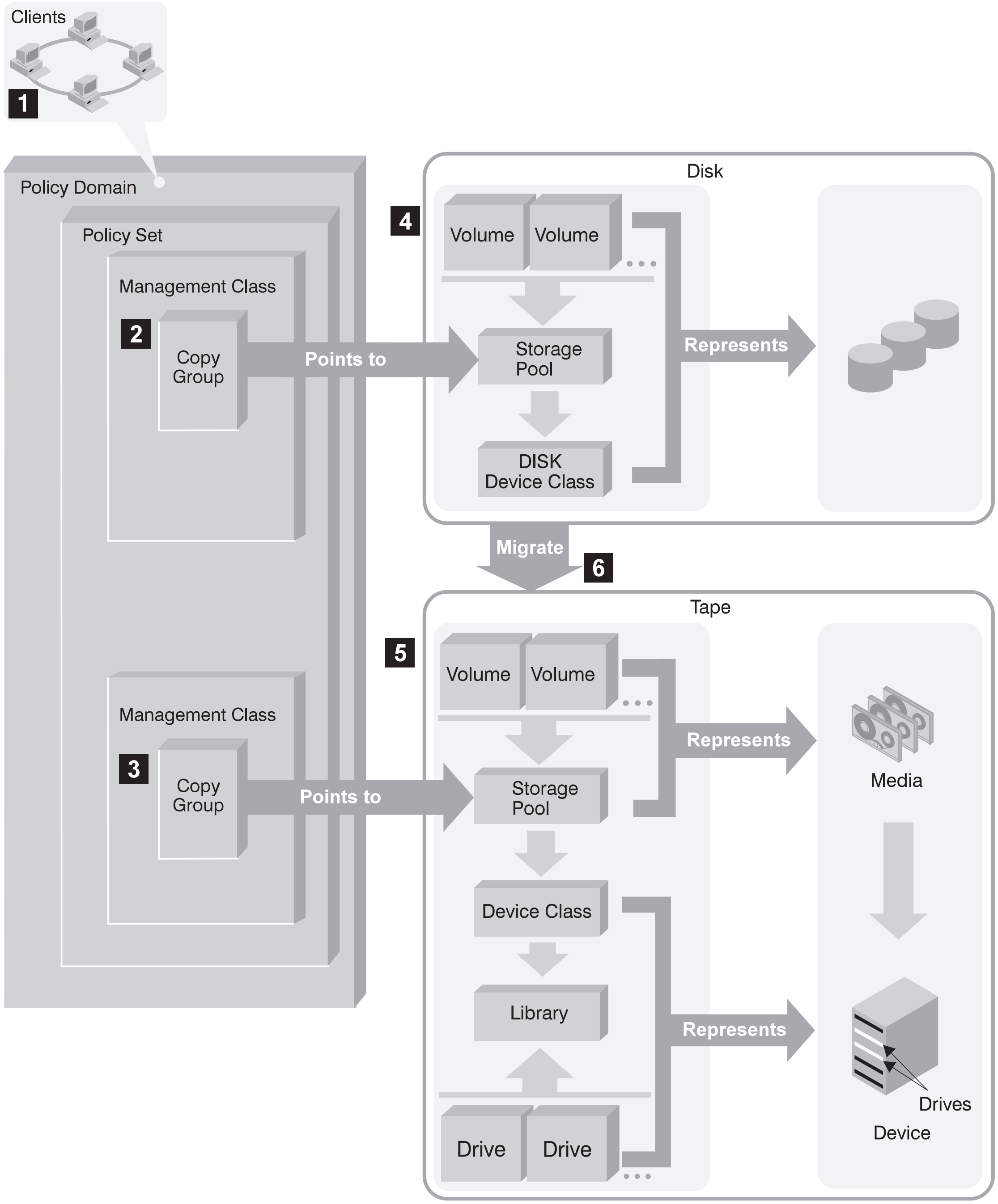


Figure 45. How Clients, Server Storage, and Policy Work Together

1 When clients are registered, they are associated with a policy domain. Within the policy domain are the policy set, management class, and copy groups.

2, **3**

When a client backs up, archives, or migrates a file, it is bound to a management class. A management class and the backup and archive copy groups within it specify where files are stored and how they are managed when they are backed up, archived, or migrated from the client.

4, **5**

Storage pools are the destinations for backed-up, archived, or space-managed files. Copy groups specify storage pools for backed-up or archived files. Management classes specify storage pools for space-managed files.

Storage pools are mapped to device classes, which represent devices. The storage pool contains volumes of the type indicated by the associated device class. For example, a storage pool that is mapped to a device class with a device type of 8MM contains only 8mm tapes.

6 Files that are initially stored on disk storage pools can migrate to tape or optical disk storage pools if the pools are set up in a storage hierarchy.

More on Management Classes

Management classes are the key connection between client files and policy.

Each client node is assigned to a single policy domain, and the client node has access only to the management classes contained in the active policy set. The management classes specify whether client files are migrated to storage pools (hierarchical storage management). The copy groups in these management classes specify the number of backup versions retained in server storage and the length of time to retain backup versions and archive copies.

For example, if a group of users needs only one backup version of their files, you can create a policy domain that contains only one management class whose backup copy group allows only one backup version. Then you can assign the client nodes for these users to the policy domain. See "Registering Nodes with the Server" on page 228 for information on registering client nodes and assigning them to policy domains.

The following sections give you more information about management classes and how they work with other parts of Tivoli Storage Manager:

- "Contents of a Management Class"
- "Default Management Classes" on page 280
- "The Include-Exclude List" on page 280
- "How Files and Directories Are Associated with a Management Class" on page 282

Contents of a Management Class

A management class contains policy for backup, archive, and space management operations by clients. You can specify if and how a Tivoli Space Manager client can migrate files to server storage with parameters in the management class. For clients using the server for backup and archive, you can choose what a management class contains from the following options:

A backup copy group and an archive copy group
Typical end users need to back up and archive documents, spreadsheets, and graphics.

A backup copy group only
Some users only want to back up files (such as working documents, database, log, or history files that change daily). Some application clients (Tivoli Data Protection products) need only a backup copy group because they never archive files.

An archive copy group only

A management class that contains only an archive copy group is useful for users who create:

- Point-in-time files. For example, an engineer can archive the design of an electronic component and the software that created the design. Later, the engineer can use the design as a base for a new electronic component.
- Files that are rarely used but need to be retained for a long time. A client can erase the original file without affecting how long the archive copy is retained in server storage. Examples include legal records, patient records, and tax forms.

Attention: A management class that contains neither a backup nor an archive copy group prevents a file from ever being backed up or archived. This type of management class is not recommended for most users. Use such a management class carefully to prevent users from mistakenly selecting it. If users bind their files to a management class without copy groups, Tivoli Storage Manager issues warning messages.

Default Management Classes

Each policy set must include a default management class, which is used for the following purposes:

- To manage files that are not bound to a specific management class, as defined by the INCLUDE option in the include-exclude list.
- To manage existing backup versions when an administrator deletes a management class or a backup copy group from the server. See "How Files and Directories Are Associated with a Management Class" on page 282.
- To manage existing archive copies when an administrator deletes a management class or an archive copy group from the server. The server does not rebind archive copies, but does use the archive copy group (if one exists) in the default management class. See "How Files and Directories Are Associated with a Management Class" on page 282.
- To manage files when a client node is assigned to a new policy domain and the active policy set does not have management classes with the same names as that to which the node's files are bound.

A typical default management class should do the following:

- Meet the needs of most users
- Contain both a backup copy group and an archive copy group
- Set serialization static or shared static to ensure the integrity of backed up and archived files
- Retain backup versions and archive copies for a sufficient amount of time
- Retain directories for at least as long as any files are associated with the directory

Other management classes can contain copy groups tailored either for the needs of special sets of users or for the needs of most users under special circumstances.

The Include-Exclude List

A user can define an include-exclude list to specify which files are eligible for the different processes that the client can run. Include and exclude options in the list determine which files are eligible for backup and archive services, which files can be migrated from the client (space-managed), and how the server manages backed-up, archived, and space-managed files.

If a user does not create an include-exclude list, the following default conditions apply:

- All files belonging to the user are eligible for backup and archive services.
- The default management class governs backup, archive, and space-management policies.

Figure 46 shows an example of an include-exclude list. The statements in this example list do the following:

- Excludes certain files or directories from backup, archive, and client migration operations

  Line 1 in Figure 46 means that the SSTEINER node ID excludes all core files from being eligible for backup and client migration.
- Includes some previously excluded files

  Line 2 in Figure 46 means that the files in the /home/ssteiner directory are excluded. The include statement that follows on line 3, however, means that the /home/ssteiner/options.scr file is eligible for backup and client migration.
- Binds a file to a specific management class

  Line 4 in Figure 46 means that all files and subdirectories belonging to the /home/ssteiner/driver5 directory are managed by the policy defined in the MCENGBK2 management class.

```
exclude /.../core
exclude /home/ssteiner/*
include /home/ssteiner/options.scr
include /home/ssteiner/driver5/.../* mcengbk2
```

Figure 46. Example of an Include-Exclude List

Tivoli Storage Manager processes the include-exclude list from the bottom up, and stops when it finds an include or exclude statement that matches the file it is processing. Therefore, the order in which the include and exclude options are listed affects which files are included and excluded. For example, suppose you switch the order of two lines in the example, as follows:

```
include /home/ssteiner/options.scr
exclude /home/ssteiner/*
```

The exclude statement comes last, and excludes all files in the /home/ssteiner directory. When Tivoli Storage Manager is processing the include-exclude list for the options.scr file, it finds the exclude statement first. This time, the options.scr file is *excluded*.

Some options are evaluated after the more basic include and exclude options. For example, options that exclude or include files for compression are evaluated after the program determines which files are eligible for the process being run.

You can create include-exclude lists as part of client options sets that you define for clients. For information on defining client option sets and assigning a client option set to a client, see “Creating Client Option Sets from the Server” on page 253.

For detailed information on the include and exclude options, see the user's guide for the appropriate client.

How Files and Directories Are Associated with a Management Class

Binding is the process of associating a file with a management class. The policies defined in the management class then apply to the bound files. The server binds a file to a management class when a client backs up, archives, or migrates the file. A client chooses a management class as follows:

- For backing up a file, a client can specify a management class in the client's include-exclude list (include-exclude options file for UNIX clients), or can accept the default management class.
- For backing up directories, the client can specify a management class by using the DIRMC option in the client options file.

  Note: It is recommended that you define a default management class. If no management class is specified for a directory, the server chooses the management class with the longest retention period in the backup copy group (retention period for the only backup version).
- For backing up a file system or logical volume, a client can specify a management class in the client's include-exclude list (include-exclude options file for UNIX clients), or can accept the default management class.
- For archiving a file, the client can do one of the following:
  - Specify a management class in the client's include-exclude list (with either an `include` option or an `include.archive` option)
  - Specify a management class with the ARCHMC option on the archive command
  - Accept the default management class
- For archiving directories, the client can specify a management class with the archiving options, or the ARCHMC option. If the client does not specify any archiving options, the server assigns the default management class to the archived directory. If the default management class has no archive copy group, the server assigns the management class that currently has the archive copy group with the shortest retention time.
- For migrating a file, a client can specify a management class in the client's include-exclude options file, or can accept the default management class.

The default management class is the management class identified as the default in the active policy set.

A management class specified with a simple `include` option can apply to one or more processes on the client. More specific include options (such as `include.archive`) allow the user to specify different management classes. Some examples of how this works:

- If a client backs up, archives, and migrates a file to the same server, and uses only a single include option, the management class specified for the file applies to all three operations (backup, archive, and migrate).
- If a client backs up and archives a file to one server, and migrates the file to a different server, the client can specify one management class for the file for backup and archive operations, and a different management class for migrating.
- Clients at Version 4.2 or later can specify a management class for archiving that is different from the management class for backup.

See the user's guide for the appropriate client for details.

Effects of Changing a Management Class

A file remains bound to a management class even if the attributes of the management class or its copy groups change. The following scenario illustrates this process:

1. A file named REPORT.TXT is bound to the default management class that contains a backup copy group specifying that up to three backup versions can be retained in server storage.
2. During the next week, three backup versions of REPORT.TXT are stored in server storage. The active and two inactive backup versions are bound to the default management class.
3. The administrator assigns a new default management class that contains a backup copy group specifying only up to two backup versions.
4. The administrator then activates the policy set, and the new default management class takes effect.
5. REPORT.TXT is backed up again, bringing the number of versions to four. The server determines that according to the new backup copy group only two versions are to be retained. Therefore, the server marks the two oldest versions for deletion (expired).
6. Expiration processing occurs (see "Running Expiration Processing to Delete Expired Files" on page 302 for details). REPORT.TXT is still bound to the default management class, which now includes new retention criteria. Therefore, the two versions marked for deletion are purged, and one active and one inactive backup version remain in storage.

Rebinding Files to Management Classes

Rebinding is the process of associating a file or a logical volume image with a new management class.

Backup Versions: The server rebinds backup versions of files and logical volume images in the following cases:

- The user changes the management class specified in the include-exclude list and does a backup.
- An administrator activates a policy set in the same policy domain as the client node, and the policy set does not contain a management class with the same name as the management class to which a file is currently bound.
- An administrator assigns a client node to a different policy domain, and the active policy set in that policy domain does not have a management class with the same name.

Backup versions of a directory can be rebound when the user specifies a different management class using the DIRMC option in the client option file, and when the directory gets backed up.

If a file is bound to a management class that no longer exists, the server uses the default management class to manage the backup versions. When the user does another backup, the server rebinds the file and any backup versions to the default management class. If the default management class does not have a backup copy group, the server uses the backup retention grace period specified for the policy domain.

Archive Copies: Archive copies are never rebound because each archive operation creates a different archive copy. Archive copies remain bound to the management class name specified when the user archived them.

If the management class to which an archive copy is bound no longer exists or no longer contains an archive copy group, the server uses the default management class. If you later change or replace the default management class, the server uses the updated default management class to manage the archive copy.

If the default management class does not contain an archive copy group, the server uses the archive retention grace period specified for the policy domain.

How Tivoli Storage Manager Selects Files for Policy Operations

This section describes how Tivoli Storage Manager selects files for the following operations:

- Full and partial incremental backups
- Selective backup
- Logical volume backup
- Archive
- Automatic migration from an HSM client (Tivoli Space Manager)

Incremental Backup

Backup-archive clients can choose to back up their files using full or partial incremental backup. A full incremental backup ensures that clients' backed-up files are always managed according to policies. Clients should use full incremental backup whenever possible.

If the amount of time for backup is limited, clients may sometimes need to use partial incremental backup. A partial incremental backup should complete more quickly and require less memory. When a client uses partial incremental backup, only files that have changed since the last incremental backup are backed up. Attributes in the management class that would cause a file to be backed up when doing a full incremental backup are ignored. For example, unchanged files are not backed up even when they are assigned to a management class that specifies absolute mode and the minimum days between backups (frequency) has passed.

The server also does less processing for a partial incremental backup. For example, the server does not expire files or rebind management classes to files during a partial incremental backup.

If clients must use partial incremental backups, they should periodically perform full incremental backups to ensure that complete backups are done and backup files are stored according to policies. For example, clients can do partial incremental backups every night during the week, and a full incremental backup on the weekend.

Performing full incremental backups is important if clients want the ability to restore files to a specific time. Only a full incremental backup can detect whether files have been deleted since the last backup. If full incremental backup is not done often enough, clients who restore to a specific time may find that many files that had actually been deleted from the workstation get restored. As a result, a client's file system may run out of space during a restore process. See "Setting Policy to Enable Point-in-Time Restore for Clients" on page 308 for more information.

Full Incremental Backup

When a user requests a full incremental backup, Tivoli Storage Manager performs the following steps to determine eligibility:

1. Checks each file against the user's include-exclude list:
   - Files that are excluded are not eligible for backup.
   - If files are not excluded and a management class is specified with the INCLUDE option, Tivoli Storage Manager uses that management class.
   - If files are not excluded but a management class is not specified with the INCLUDE option, Tivoli Storage Manager uses the default management class.
   - If no include-exclude list exists, all files in the client domain are eligible for backup, and Tivoli Storage Manager uses the default management class.
2. Checks the management class of each included file:
   - If there is a backup copy group, the process continues with step 3.
   - If there is no backup copy group, the file is not eligible for backup.
3. Checks the *mode*, *frequency*, and *serialization* defined in the backup copy group.

   Mode Specifies whether the file is backed up only if it has changed since the last backup (*modified*) or whenever a backup is requested (*absolute*).

   Frequency
   Specifies the minimum number of days that must elapse between backups.

   Note: For Windows NT and Windows 2000 this attribute is ignored during a journal-based backup.

   Serialization
   Specifies how files are handled if they are modified while being backed up and what happens if modification occurs.

   - If the mode is *modified* and the minimum number of days have elapsed since the file was last backed up, Tivoli Storage Manager determines if the file has been changed since it was last backed up:
     – If the file has been changed and the serialization requirement is met, the file is backed up.
     – If the file has not been changed, it is not backed up.
   - If the mode is *modified* and the minimum number of days have not elapsed since the file was last backed up, the file is not eligible for backup.
   - If the mode is *absolute*, the minimum number of days have elapsed since the file was last backed up, and the serialization requirement is met, the file is backed up.
   - If the mode is *absolute* and the minimum number of days have not elapsed since the file was last backed up, the file is not eligible for backup.

Partial Incremental Backup

When a user requests a partial incremental backup, Tivoli Storage Manager performs the following steps to determine eligibility:

1. Checks each file against the user's include-exclude list:
   - Files that are excluded are not eligible for backup.
   - If files are not excluded and a management class is specified with the INCLUDE option, Tivoli Storage Manager uses that management class.

- If files are not excluded but a management class is not specified with the INCLUDE option, Tivoli Storage Manager uses the default management class.
- If no include-exclude list exists, all files in the client domain are eligible for backup, and Tivoli Storage Manager uses the default management class.

2. Checks the management class of each included file:
   - If there is a backup copy group, the process continues with step 3.
   - If there is no backup copy group, the file is not eligible for backup.
3. Checks the date and time of the last incremental backup by the client, and the *serialization* requirement defined in the backup copy group. (Serialization specifies how files are handled if they are modified while being backed up and what happens if modification occurs.)
   - If the file has not changed since the last incremental backup, the file is not backed up.
   - If the file has changed since the last incremental backup and the serialization requirement is met, the file is backed up.

Selective Backup

When a user requests a selective backup, Tivoli Storage Manager performs the following steps to determine eligibility:

1. Checks the file against any include or exclude statements contained in the user include-exclude list:
   - Files that are not excluded are eligible for backup. If a management class is specified with the INCLUDE option, Tivoli Storage Manager uses that management class.
   - If no include-exclude list exists, the files selected are eligible for backup, and Tivoli Storage Manager uses the default management class.
2. Checks the management class of each included file:
   - If the management class contains a backup copy group and the serialization requirement is met, the file is backed up. Serialization specifies how files are handled if they are modified while being backed up and what happens if modification occurs.
   - If the management class does not contain a backup copy group, the file is not eligible for backup.

An important characteristic of selective backup is that a file is backed up without regard for whether the file has changed. This result may not always be what you want. For example, suppose a management class specifies to keep three backup versions of a file. If the client uses incremental backup, the file is backed up only when it changes, and the three versions in storage will be at different levels. If the client uses selective backup, the file is backed up regardless of whether it has changed. If the client uses selective backup on the file three times without changing the file, the three versions of the file in server storage are identical. Earlier, different versions are lost.

Logical Volume Backup

When a user requests a logical volume backup, Tivoli Storage Manager performs the following steps to determine eligibility:

1. Checks the specification of the logical volume against any include or exclude statements contained in the user include-exclude list:

- If no include-exclude list exists, the logical volumes selected are eligible for backup, and Tivoli Storage Manager uses the default management class.
- Logical volumes that are not excluded are eligible for backup. If the include-exclude list has an INCLUDE option for the volume with a management class specified, Tivoli Storage Manager uses that management class. Otherwise, the default management class is used.

2. Checks the management class of each included logical volume:
   - If the management class contains a backup copy group and the logical volume meets the serialization requirement, the logical volume is backed up. Serialization specifies how logical volumes are handled if they are modified while being backed up and what happens if modification occurs.
   - If the management class does not contain a backup copy group, the logical volume is not eligible for backup.

Archive

When a user requests the archiving of a file or a group of files, Tivoli Storage Manager performs the following steps to determine eligibility:

1. Checks the files against the user's include-exclude list to see if any management classes are specified:
   - Tivoli Storage Manager uses the default management class for files that are not bound to a management class.
   - If no include-exclude list exists, Tivoli Storage Manager uses the default management class unless the user specifies another management class. See the user's guide for the appropriate client for details.
2. Checks the management class for each file to be archived.
   - If the management class contains an archive copy group and the serialization requirement is met, the file is archived. Serialization specifies how files are handled if they are modified while being archived and what happens if modification occurs.
   - If the management class does not contain an archive copy group, the file is not archived.

Note: If you need to frequently create archives for the same data, consider using instant archive (backup sets) instead. Frequent archive operations can create a large amount of metadata in the server database resulting in increased database growth and decreased performance for server operations such as expiration. Frequently, you can achieve the same objectives with incremental backup or backup sets. Although the archive function is a powerful way to store inactive data with fixed retention, it should not be used on a frequent and large scale basis as the primary backup method. For details on how to generate backup sets see "Creating and Using Client Backup Sets" on page 316.

Automatic Migration from a Client Node

A file is eligible for automatic migration from an HSM client if it meets all of the following criteria:

- It resides on a node on which the root user has added and activated hierarchical storage management. It must also reside in a local file system to which the root user has added space management, and not in the root (/) or /tmp file system.
- It is not excluded from migration in the include-exclude list.
- It meets management class requirements for migration:

- The file is not a character special file, a block special file, a FIFO special file (that is, a named pipe file) or a directory.
- The file is assigned to a management class that calls for space management.
- The management class calls for automatic migration after a specified number of days, and that time has elapsed.
- A backup version of the file exists if the management class requires it.
- The file is larger than the stub file that would replace it (plus one byte) or the file system block size, whichever is larger.

How Client Migration Works with Backup and Archive

As an administrator, you can define a management class that specifies automatic migration from the client under certain conditions. For example, if the file has not been accessed for at least 30 days and a backup version exists, the file is migrated. You can also define a management class that allows users to selectively migrate whether or not a backup version exists. Users can also choose to archive files that have been migrated. Tivoli Storage Manager does the following:

- If the file is backed up or archived to the server to which it was migrated, the server copies the file from the migration storage pool to the backup or archive storage pool. For a tape-to-tape operation, each storage pool must have a tape drive.
- If the file is backed up or archived to a different server, Tivoli Storage Manager accesses the file by using the migrate-on-close recall mode. The file resides on the client node only until the server stores the backup version or the archived copy in a storage pool.

When a client restores a backup version of a migrated file, the server deletes the migrated copy of the file from server storage the next time reconciliation is run.

When a client archives a file that is migrated and does not specify that the file is to be erased after it is archived, the migrated copy of the file remains in server storage. When a client archives a file that is migrated and specifies that the file is to be erased, the server deletes the migrated file from server storage the next time reconciliation is run.

Tivoli Storage Manager's default management class specifies that a backup version of a file must exist before the file is eligible for migration.

Creating Your Own Policies

Task	Required Privilege Class
Define or copy a policy domain	System
Update a policy domain over which you have authority	Restricted policy
Define, update, or copy policy sets and management classes in any policy domain	System or unrestricted policy
Define, update, or copy policy sets and management classes in policy domains over which you have authority	Restricted policy
Define or update copy groups in any policy domain	System or unrestricted policy

Task	Required Privilege Class
Define or update copy groups that belong to policy domains over which you have authority	Restricted policy
Assign a default management class to a nonactive policy set in any policy domain	System or unrestricted policy
Assign a default management class to a nonactive policy set in policy domains over which you have authority	Restricted policy
Validate and activate policy sets in any policy domain	System or unrestricted policy
Validate and activate policy sets in policy domains over which you have authority	Restricted policy
Start inventory expiration processing	System

You can create your own policies in one of two ways:

- Define the parts of a policy and specify each attribute
- Copy existing policy parts and update only those attributes that you want to change

The following table shows that an advantage of copying existing policy parts is that some associated parts are copied in a single operation.

If you copy this...	Then you create this...
Policy Domain	A new policy domain with: • A copy of each policy set from the original domain • A copy of each management class in each original policy set • A copy of each copy group in each original management class
Policy Set	A new policy set *in the same policy domain* with: • A copy of each management class in the original policy set • A copy of each copy group in the original management class
Management Class	A new management class *in the same policy set* and a copy of each copy group in the management class

Example: Sample Policy Objects

Figure 47 on page 290 shows the policies for an engineering department. This example is used throughout the rest of this chapter.

The domain contains two policy sets that are named STANDARD and TEST. The administrator activated the policy set that is named STANDARD. When you activate a policy set, the server makes a copy of the policy set and names it ACTIVE. Only one policy set can be active at a time.

The ACTIVE policy set contains two management classes: MCENG and STANDARD. The default management class is STANDARD.

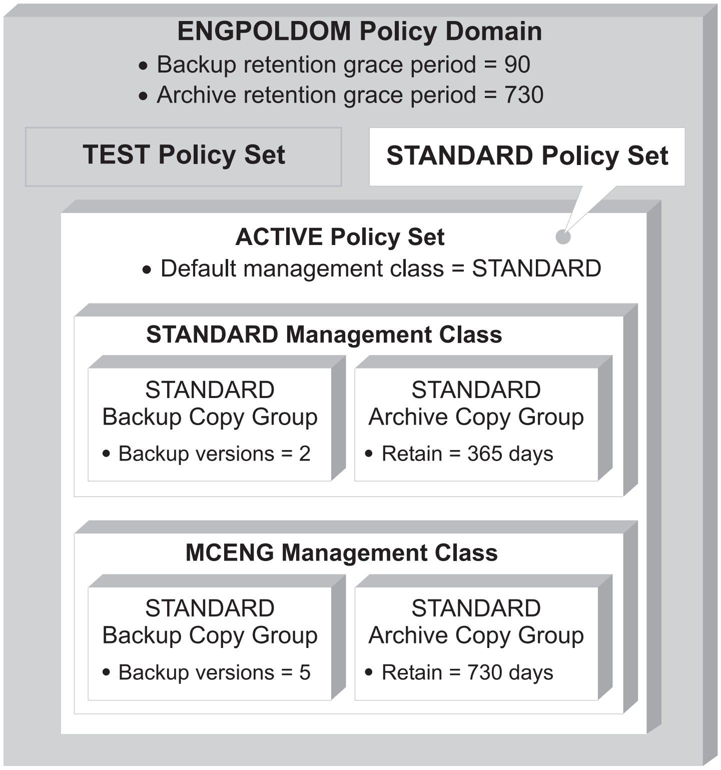

ENGPOLDOM Policy Domain
• Backup retention grace period = 90
• Archive retention grace period = 730
TEST Policy Set
STANDARD Policy Set
ACTIVE Policy Set
• Default management class = STANDARD
STANDARD Management Class
STANDARD Backup Copy Group
• Backup versions = 2
STANDARD Archive Copy Group
• Retain = 365 days
MCENG Management Class
STANDARD Backup Copy Group
• Backup versions = 5
STANDARD Archive Copy Group
• Retain = 730 days

Figure 47. An Example of Policy Objects Defined for an Engineering Department

The sections that follow describe the tasks involved in creating new policies for your installation. Do the tasks in the following order:

Tasks:
"Defining and Updating a Policy Domain"
"Defining and Updating a Policy Set" on page 291
"Defining and Updating a Management Class" on page 292
"Defining and Updating a Backup Copy Group" on page 293
"Defining and Updating an Archive Copy Group" on page 298
"Assigning a Default Management Class" on page 300
"Activating a Policy Set" on page 301
"Running Expiration Processing to Delete Expired Files" on page 302.

Defining and Updating a Policy Domain

When you update or define a policy domain, you specify:

Backup Retention Grace Period

Specifies the number of days to retain an inactive backup version when the server cannot rebind the file to an appropriate management class. The backup retention grace period protects backup versions from being immediately expired when the management class to which a file is bound no longer exists or no longer contains a backup copy group, and the default management class does not contain a backup copy group.

Backup versions of the file managed by the grace period are retained in server storage only for the backup retention grace period. This period starts from the day of the backup. For example, if the backup retention grace period for the STANDARD policy domain is used and set to 30 days, backup versions using the grace period expire in 30 days from the day of the backup.

Backup versions of the file continue to be managed by the grace period unless one of the following occurs:

- The client binds the file to a management class containing a backup copy group and then backs up the file
- A backup copy group is added to the file's management class
- A backup copy group is added to the default management class

Archive Retention Grace Period

Specifies the number of days to retain an archive copy when the management class for the file no longer contains an archive copy group and the default management class does not contain an archive copy group. The retention grace period protects archive copies from being immediately expired.

The archive copy of the file managed by the grace period is retained in server storage for the number of days specified by the archive retention grace period. This period starts from the day on which the file is first archived. For example, if the archive retention grace period for the policy domain STANDARD is used, an archive copy expires 365 days from the day the file is first archived.

The archive copy of the file continues to be managed by the grace period unless an archive copy group is added to the file's management class or to the default management class.

Example: Defining a Policy Domain

To create a new policy domain you can do one of the following:

- Copy an existing policy domain and update the new domain
- Define a new policy domain from scratch

Note: When you copy an existing domain, you also copy any associated policy sets, management classes, and copy groups.

For example, to copy and update, follow this procedure:

1. Copy the STANDARD policy domain to the ENGPOLDOM policy domain by entering:

   ```
   copy domain standard engpoldom
   ```

   ENGPOLDOM now contains the standard policy set, management class, backup copy group, and archive copy group.
2. Update the policy domain ENGPOLDOM so that the backup retention grace period is extended to 90 days and the archive retention grace period is extended to 2 years by entering:

   ```
   update domain engpoldom description='Engineering Policy Domain'
   backretention=90 archretention=730
   ```

Defining and Updating a Policy Set

When you define or update a policy set, specify:

Policy domain name

Names the policy domain to which the policy set belongs

The policies in the new policy set do not take effect unless you make the new set the ACTIVE policy set. See "Activating a Policy Set" on page 301.

Example: Defining a Policy Set

An administrator needs to develop new policies based on the existing STANDARD policy set. To create the TEST policy set in the ENGPOLDOM policy domain, the administrator performs the following steps:

1. Copy the STANDARD policy set and name the new policy set TEST:

   ```
   copy policyset engpoldom standard test
   ```

   Note: When you copy an existing policy set, you also copy any associated management classes and copy groups.
2. Update the description of the policy set named TEST:

   ```
   update policyset engpoldom test
   description='Policy set for testing'
   ```

Defining and Updating a Management Class

When you define or update a management class, specify:

Policy domain name
Names the policy domain to which the management class belongs.

Policy set name
Names the policy set to which the management class is assigned.

Description
Describes the management class. A clear description can help users to choose an appropriate management class for their use.

The following four parameters apply only to Tivoli Space Manager clients (HSM clients):

Whether space management is allowed
Specifies that the files are eligible for both automatic and selective migration, only selective migration, or no migration.

How frequently files can be migrated
Specifies the minimum number of days that must elapse since a file was last accessed before it is eligible for automatic migration.

Whether backup is required
Specifies whether a backup version of a file must exist before the file can be migrated.

Where migrated files are to be stored
Specifies the name of the storage pool in which migrated files are stored. Your choice could depend on factors such as:

- The number of client nodes migrating to the storage pool. When many user files are stored in the same storage pool, volume contention can occur as users try to migrate files to or recall files from the storage pool.
- How quickly the files must be recalled. If users need immediate access to migrated versions, you can specify a disk storage pool as the destination.

Note: You cannot specify a copy storage pool as a destination.

Example: Define a New Management Class

Create a new management class by following these steps:

1. Define a new management class MCENG by entering:

   ```
   define mgmtclass engpoldom standard mceng
   ```

2. Update the description of the MCENG management class by entering:

   ```
   update mgmtclass engpoldom standard mceng
   description='Engineering Management Class for Backup and Archive'
   ```

Defining and Updating a Backup Copy Group

Tasks:
"Where to Store Backed-Up Files"
"How to Handle Files That Are Modified During Backup"
"How Frequently Files Can Be Backed Up" on page 294
"How Many Backup Versions to Retain and For How Long" on page 295

Where to Store Backed-Up Files

Specify a storage pool where the server initially stores the files associated with this backup copy group. This is called the destination. Your choice can depend on factors such as the following:

- Whether the server and the client nodes have access to shared devices on a storage area network (SAN).
- The number of client nodes backing up to the storage pool. When many user files are stored in the same storage pool, volume contention can occur as users try to back up to or restore files from the storage pool.
- How quickly the files must be restored. If users need immediate access to backup versions, you may want to specify a disk storage pool as the destination.

Note: You cannot specify a copy storage pool.

How to Handle Files That Are Modified During Backup

You can use the SERIALIZATION attribute on the DEFINE COPYGROUP command to specify how files are handled if they are modified during a backup. This attribute can be one of four values: STATIC, SHRSTATIC (shared static), DYNAMIC, or SHRDYNAMIC (shared dynamic). The value you choose depends on how you want Tivoli Storage Manager to handle files that are modified while they are being backed up.

Do not back up files that are modified during the backup

You will want to prevent the server from backing up a file while it is being modified. Use one of the following values:

STATIC

Specifies that Tivoli Storage Manager will attempt to back up the file only once. If the file or directory is modified during a backup, the server does not back it up.

SHRSTATIC (Shared static)

Specifies that if the file or directory is modified during a backup, the server retries the backup as many times as specified by the CHANGINGRETRIES option in the client options file. If the file is modified during the last attempt, the file or directory is not backed up.

Back Up Files That Are Modified During the Backup

Some files are in constant use, such as an error log. Consequently, these files may never be backed up when serialization is set to STATIC or SHRSTATIC. To back up files that are modified during the backup, use one of the following values:

DYNAMIC

Specifies that a file or directory is backed up on the first attempt, even if the file or directory is modified during the backup.

SHRDYNAMIC (Shared dynamic)

Specifies that if a file or directory is modified during a backup, the server retries the backup as many times as specified by the CHANGINGRETRIES option in the client options file. The server backs up the file on the last attempt, even if the file or directory is being modified.

Attention:

- If a file is modified during backup and DYNAMIC or SHRDYNAMIC is specified, then the backup may not contain all the changes and may not be usuable. For example, the backup version may contain a truncated record. Under some circumstances, it may be acceptable to capture a dynamic or ″fuzzy″ backup of a file (the file was changed during the backup). For example, a dynamic backup of an error log file that is continuously appended may be acceptable. However, a dynamic backup of a database file may not be acceptable, since restoring such a backup could result in an unusable database. Carefully consider dynamic backups of files as well as possible problems that may result from restoring potentially ″fuzzy″ backups.
- When certain users or processes open files, they may deny any other access, including ″read″ access, to the files by any other user or process. When this happens, even with serialization set to DYNAMIC or SHRDYNAMIC, Tivoli Storage Manager will not be able to open the file at all, so the server cannot back up the file.

How Frequently Files Can Be Backed Up

You can specify how frequently files can be backed up with two parameters:

Frequency

The frequency is the minimum number of days that must elapse between full incremental backups.

Note: For Windows NT and Windows 2000 this attribute is ignored during a journal-based backup.

Mode The mode parameter specifies whether a file or directory must have been modified to be considered for backup during a full incremental backup process. Tivoli Storage Manager does not check this attribute when a user requests a partial incremental backup, a selective backup for a file, or a backup of a logical volume. You can select from two modes:

Modified

A file is considered for full incremental backup only if it has changed since the last backup. A file is considered changed if any of the following items is different:

- Date on which the file was last modified
- File size
- File owner

- File permissions

Absolute
A file is considered for full incremental backup regardless of whether it has changed since the last backup.

The server considers both parameters to determine how frequently files can be backed up. For example, if frequency is 3 and mode is Modified, a file or directory is backed up only if it has been changed and if three days have passed since the last backup. If frequency is 3 and mode is Absolute, a file or directory is backed up after three days have passed whether or not the file has changed.

Use the Modified mode when you want to ensure that the server retains multiple, *different* backup versions. If you set the mode to Absolute, users may find that they have three *identical* backup versions, rather than three *different* backup versions.

Absolute mode can be useful for forcing a full backup. It can also be useful for ensuring that extended attribute files are backed up, because TSM does not detect changes if the size of the extended attribute file remains the same.

When you set the mode to Absolute, set the frequency to 0 if you want to ensure that a file is backed up each time full incremental backups are scheduled for or initiated by a client.

How Many Backup Versions to Retain and For How Long

Multiple versions of files are useful when users continually update files and sometimes need to restore the original file from which they started. The most current backup version of a file is called the *active* version. All other versions are called *inactive* versions. You can specify the number of versions to keep by:

- Directly specifying the number of versions

  You specify the number of backup versions with two parameters:
  - Versions Data Exists (number of versions to keep when the data still exists on the client node)
  - Versions Data Deleted (number of versions to keep when the data no longer exists on the client node)
- Specifying the number of days to keep each backup version

  You specify the number of days to keep backup versions with two parameters:
  - Retain Extra Versions (how many days to keep inactive backup versions; the days are counted from the day that the version became inactive)
  - Retain Only Versions (how many days to keep the last backup version of a file that has been deleted)
- Specifying a combination of the number of versions and the days to keep them

  Use a combination of the four parameters: Versions Data Exists, Versions Data Deleted, Retain Extra Versions, and Retain Only Versions.

These parameters interact to determine the backup versions that the server retains. When the number of inactive backup versions exceeds the number of versions allowed (Versions Data Exists and Versions Data Deleted), the oldest version expires and the server deletes the file from the database the next time expiration processing runs. How many inactive versions the server keeps is also related to the parameter for how long inactive versions are kept (Retain Extra Versions). Inactive versions expire when the number of days that they have been inactive exceeds the value specified for retaining extra versions, even when the number of versions is not exceeded.

Note: A base file is not eligible for expiration until all its dependent subfiles have been expired. For details, see "Enabling Clients to Use Subfile Backup" on page 322.

For example, see Table 23 and Figure 48. A client node has backed up the file REPORT.TXT four times in one month, from March 23 to April 23. The settings in the backup copy group of the management class to which REPORT.TXT is bound determine how the server treats these backup versions. Table 24 shows some examples of how different copy group settings would affect the versions. The examples show the effects as of April 24 (one day after the file was last backed up).

Table 23. Status of REPORT.TXT as of April 24

Version	Date Created	Days the Version Has Been Inactive
Active	April 23	(not applicable)
Inactive 1	April 13	1 (since April 23)
Inactive 2	March 31	11 (since April 13)
Inactive 3	March 23	24 (since March 31)

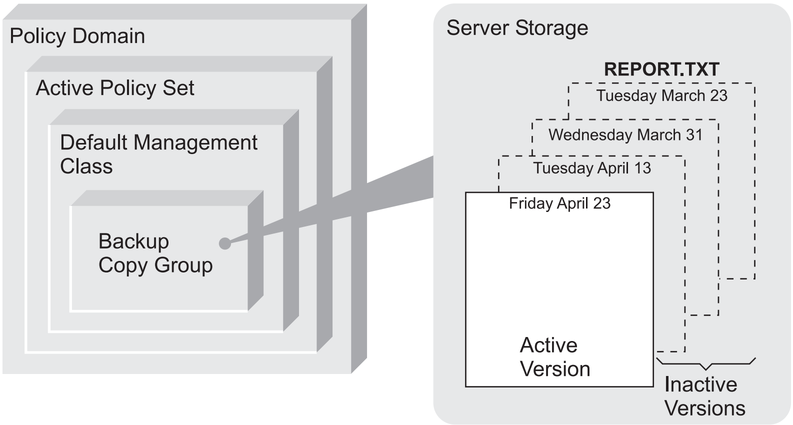


Figure 48. Active and Inactive Versions of REPORT.TXT

Table 24. Effects of Backup Copy Group Policy on Backup Versions for REPORT.TXT as of April 24. One day after the file was last backed up.

Versions Data Exists	Versions Data Deleted	Retain Extra Versions	Retain Only Version	Results
3 versions	2 versions	60 days	180 days	Versions Data Exists and Retain Extra Versions control the expiration of the versions. The version created on March 23 is retained until the client node backs up the file again (creating a fourth inactive version), or until that version has been inactive for 60 days. If the user deletes the REPORT.TXT file from the client node, the server notes the deletion at the next full incremental backup of the client node. From that point, the Versions Data Deleted and Retain Only Version parameters also have an effect. All versions are now inactive. Two of the four versions expire immediately (the March 23 and March 31 versions expire). The April 13 version expires when it has been inactive for 60 days (on June 23). The server keeps the last remaining inactive version, the April 23 version, for 180 days after it becomes inactive.

Table 24. Effects of Backup Copy Group Policy on Backup Versions for REPORT.TXT as of April 24 (continued). One day after the file was last backed up.

Versions Data Exists	Versions Data Deleted	Retain Extra Versions	Retain Only Version	Results
NOLIMIT	2 versions	60 days	180 days	Retain Extra Versions controls expiration of the versions. The inactive versions (other than the last remaining version) are expired when they have been inactive for 60 days. If the user deletes the REPORT.TXT file from the client node, the server notes the deletion at the next full incremental backup of the client node. From that point, the Versions Data Deleted and Retain Only Version parameters also have an effect. All versions are now inactive. Two of the four versions expire immediately (the March 23 and March 31 versions expire) because only two versions are allowed. The April 13 version expires when it has been inactive for 60 days (on June 22). The server keeps the last remaining inactive version, the April 23 version, for 180 days after it becomes inactive.
NOLIMIT	NOLIMIT	60 days	180 days	Retain Extra Versions controls expiration of the versions. The server does not expire inactive versions based on the maximum number of backup copies. The inactive versions (other than the last remaining version) are expired when they have been inactive for 60 days. If the user deletes the REPORT.TXT file from the client node, the server notes the deletion at the next full incremental backup of the client node. From that point, the Retain Only Version parameter also has an effect. All versions are now inactive. The three of four versions will expire after each of them has been inactive for 60 days. The server keeps the last remaining inactive version, the April 23 version, for 180 days after it becomes inactive.
3 versions	2 versions	NOLIMIT	NOLIMIT	Versions Data Exists controls the expiration of the versions until a user deletes the file from the client node. The server does not expire inactive versions based on age. If the user deletes the REPORT.TXT file from the client node, the server notes the deletion at the next full incremental backup of the client node. From that point, the Versions Data Deleted parameter controls expiration. All versions are now inactive. Two of the four versions expire immediately (the March 23 and March 31 versions expire) because only two versions are allowed. The server keeps the two remaining inactive versions indefinitely.

See *Administrator's Reference* for details about the parameters. The following list gives some tips on using the NOLIMIT value:

Versions Data Exists
Setting the value to NOLIMIT may require increased storage, but that value may be needed for some situations. For example, to enable client nodes to restore files to a specific point in time, set the value for Versions Data Exists to NOLIMIT. Setting the value this high ensures that the server retains versions according to the Retain Extra Versions parameter for the copy group. See "Setting Policy to Enable Point-in-Time Restore for Clients" on page 308 and "Policy for Logical Volume Backups" on page 304 for more information.

Versions Data Deleted
Setting the value to NOLIMIT may require increased storage, but that value may be needed for some situations. For example, set the value for Versions Data Deleted to NOLIMIT to enable client nodes to restore files to a specific point in time. Setting the value this high ensures that the server retains versions according to the Retain Extra Versions parameter for the

copy group. See “Setting Policy to Enable Point-in-Time Restore for Clients” on page 308 and “Policy for Logical Volume Backups” on page 304 for more information.

Retain Extra Versions

If NOLIMIT is specified, inactive backup versions are deleted based on the Versions Data Exists or Versions Data Deleted parameters.

To enable client nodes to restore files to a specific point in time, set the parameters Versions Data Exists or Versions Data Deleted to NOLIMIT. Set the value for Retain Extra Versions to the number of days that you expect clients may need versions of files available for possible point-in-time restoration. For example, to enable clients to restore files from a point in time 60 days in the past, set Retain Extra Versions to 60. See “Setting Policy to Enable Point-in-Time Restore for Clients” on page 308 for more information.

Retain Only Version

If NOLIMIT is specified, the last version is retained forever unless a user or administrator deletes the file from server storage.

Example: Define a Backup Copy Group

Define a backup copy group belonging to the MCENG management class in the STANDARD policy set belonging to the ENGPOLDOM policy domain. This new copy group must do the following:

- Let users back up changed files, regardless of how much time has elapsed since the last backup, using the default value 0 for the Frequency parameter (`frequency` parameter not specified)
- Retain up to four inactive backup versions when the original file resides on the user workstation, using the Versions Data Exists parameter (`verexists=5`)
- Retain up to four inactive backup versions when the original file is deleted from the user workstation, using the Versions Data Deleted parameter (`verdeleted=4`)
- Retain inactive backup versions for no more than 90 days, using the Retain Extra Versions parameter (`retextra=90`)
- If there is only one backup version, retain it for 600 days after the original is deleted from the workstation, using the Retain Only Version parameter (`retonly=600`)
- Prevent files from being backed up if they are in use, using the Serialization parameter (`serialization=static`)
- Store files in the ENGBACK1 storage pool, using the Destination parameter (`destination=engback1`)

To define the backup copy group, enter:

```
define copygroup engpoldom standard mceng standard
destination=engback1 serialization=static
verexists=5 verdeleted=4 retextra=90 retonly=600
```

Defining and Updating an Archive Copy Group

To define or update an archive copy group on the Web interface or command line, specify:

Where archived files are to be stored

Specify a defined storage pool as the initial destination. Your choice can depend on factors such as:

- Whether the server and the client nodes have access to shared devices on a SAN
- The number of client nodes archiving files to the storage pool. When many user files are stored in the same storage pool, volume contention can occur as users archive files to and retrieve files from the storage pool.
- How quickly the files must be restored. If users need immediate access to archive copies, you could specify a disk storage pool as the destination.
- Whether the archive copy group is for a management class that is the default for a policy domain. The default management class is used by clients registered in the policy domain, when they do not specify a management class for a file. This includes servers that are registered as clients to this server. See "Using Virtual Volumes to Store Data on Another Server" on page 469 for information about registering servers as clients to another server.

Note: You cannot specify a copy storage pool as a destination.

If files can be modified during archive

Specify how files are handled if they are modified while being archived. This attribute, called serialization, can be one of four values:

Static Specifies that if the file is modified during an archiving process, the server does not archive it. Tivoli Storage Manager does not retry the archive.

Shared Static

Specifies that if the file is modified during an archive process, the server does not archive it. However, Tivoli Storage Manager retries the archive process as many times as specified by the CHANGINGRETRIES option in the client options file.

Dynamic

Specifies that a file is archived on the first attempt, even if the file is being modified during the archive process.

Shared Dynamic

Specifies that if the file is modified during the archive attempt, the server archives it on its last try even if the file is being modified. Tivoli Storage Manager retries the archive process as many times as specified by the CHANGINGRETRIES option in the client options file.

For most files, set serialization to either static or shared static to prevent the server from archiving a file while it is being modified.

However, you may want to define a copy group with a serialization of shared dynamic or dynamic for files where log records are continuously added, such as an error log. If you only have copy groups that use static or shared static, these files may never be archived because they are constantly in use. With shared dynamic or dynamic, the log files are archived. However, the archive copy may contain a truncated message.

Attention: If a file is archived while it is in use (shared dynamic or dynamic serialization), the copy may not contain all the changes and may not be usable.

Note: When certain users or processes open files, they deny read access to the files for any other user or process. When this happens, even with serialization set to dynamic or shared dynamic, the server does not back up the file.

How long to retain an archived copy

Specifies the number of days to retain an archived copy in storage. When the time elapses, the archived copy expires and the server deletes the file the next time expiration processing runs.

When a user archives directories, the server uses the default management class unless the user specifies otherwise. If the default management class does not have an archive copy group, the server binds the directory to the management class that currently has the shortest retention time for archive. When you change the retention time for an archive copy group, you may also be changing the retention time for any directories that were archived using that copy group.

The user can change the archive characteristics by using Archive Options in the interface or by using the ARCHMC option on the command.

Example: Define an Archive Copy Group

Define an archive copy group belonging to the MCENG class that:

- Allows users to archive a file if it is not in use (`serialization=static`)
- Retains the archive copy for 730 days (`retver=730`)
- Stores files in the ENGARCH1 storage pool (`destination=engarch1`)

To define a STANDARD archive copy group to the MCENG management class in the STANDARD policy set belonging to the ENGPOLDOM policy domain, enter:

```
define copygroup engpoldom standard mceng standard
type=archive destination=engarch1 serialization=static
retver=730
```

Assigning a Default Management Class

After you have defined a policy set and the management classes that it contains, you must assign a default management class for the policy set. See “Default Management Classes” on page 280 for suggestions about the content of default management classes.

Example: Assign a Default Management Class

To assign the STANDARD management class as the default management class for the TEST policy set in the ENGPOLDOM policy domain, enter:

```
assign defmgmtclass engpoldom standard standard
```

The STANDARD management class was copied from the STANDARD policy set to the TEST policy set (see “Example: Defining a Policy Set” on page 292). Before the new default management class takes effect, you must activate the policy set.

Validating and Activating a Policy Set

After you have defined a policy set and defined management classes to it, you can validate the policy set and activate the policy set for the policy domain. Only one policy set is active in a policy domain.

Validating a Policy Set

When you validate a policy set, the server examines the management class and copy group definitions in the policy set and reports on conditions that need to be considered if the policy set is activated.

Validation fails if the policy set does not contain a default management class. Validation results in result in warning messages if any of the following conditions exist.

Condition	Reason for warning
The storage destinations specified for backup, archive, or migration do not refer to defined storage pools.	A backup, archive, or migration operation will fail when the operation involves storing a file in a storage pool that does not exist.
A storage destination specified for backup, archive, or migration is a copy storage pool.	The storage destination must be a primary storage pool.
The default management class does not contain a backup or archive copy group.	When the default management class does not contain a backup or archive copy group, any user files bound to the default management class *are not* backed up or archived.
The current ACTIVE policy set names a management class that is not defined in the policy set being validated.	When users back up files that were bound to a management class that no longer exists in the active policy set, backup versions are rebound to the default management class. See "How Files and Directories Are Associated with a Management Class" on page 282 for details. When the management class to which an archive copy is bound no longer exists and the default management class does not contain an archive copy group, the archive retention grace period is used to retain the archive copy. See "Defining and Updating a Policy Domain" on page 290 for details.
The current ACTIVE policy set contains copy groups that are not defined in the policy set being validated.	When users perform a backup and the backup copy group no longer exists in the management class to which a file is bound, backup versions are managed by the default management class. If the default management class does not contain a backup copy group, backup versions are managed by the backup retention grace period, and the workstation file is not backed up. See "Defining and Updating a Policy Domain" on page 290
A management class specifies that a backup version must exist before a file can be migrated from a client node, but the management class does not contain a backup copy group.	The contradictions within the management classes can cause problems for HSM users.

Activating a Policy Set

To activate a policy set, specify a policy domain and policy set name. When you activate a policy set, the server:

- Performs a final validation of the contents of the policy set
- Copies the original policy set to the ACTIVE policy set

You cannot update the ACTIVE policy set; the original and the ACTIVE policy sets are two separate objects. For example, updating the original policy set has no effect on the ACTIVE policy set. To change the contents of the ACTIVE policy set, you must create or change another policy set and then activate that policy set. See "Changing Policy" on page 272 for details.

Example: Validating and Activating a Policy Set

Validating and activating the STANDARD policy set in the ENGPOLDOM policy domain is a two-step process:

1. To validate the STANDARD policy set, enter:

   ```
   validate policyset engpoldom standard
   ```

   Examine any messages that result and correct the problems.
2. To activate the STANDARD policy set, enter:

   ```
   activate policyset engpoldom standard
   ```

Assigning Client Nodes to a Policy Domain

At the server command line or the administrative Web interface, you can assign existing client nodes to a new policy domain, or create new client nodes to be associated with an existing policy domain.

For example, to assign the client node APPCLIENT1 to the ENGPOLDOM policy domain, enter the following command:

```
update node appclient1 domain=engpoldom
```

To create a new client node, NEWUSER, and assign it to the ENGPOLDOM policy domain, enter the following command:

```
register node newuser newuser domain=engpoldom
```

Running Expiration Processing to Delete Expired Files

Expiration processing deletes expired client files from the server storage. Expiration processing also removes from the database any restartable restore sessions that exceed the time limit for saving such sessions.

You can run expiration processing either automatically or by command. You should ensure that expiration processing runs periodically to allow the server to reuse storage pool space that is occupied by expired client files.

Note: A base file is not eligible for expiration until all of its dependent subfiles have been expired. For details, see “Expiration Processing of Base Files and Subfiles” on page 324.

Running Expiration Processing Automatically

You control automatic expiration processing by using the expiration interval option (EXPINTERVAL) in the server options file (dsmserv.opt). You can also control when restartable restore sessions expire with another server option, RESTOREINTERVAL. You can set the options by editing the dsmserv.opt file (see *Administrator’s Reference*).

If you use the server options file to control automatic expiration, the server runs expiration processing each time you start the server. After that, the server runs expiration processing at the interval you specified with the option, measured from the start time of the server.

Using Commands and Scheduling to Control Expiration Processing

You can manually start expiration processing by issuing the following command:

```
expire inventory
```

Expiration processing then deletes expired files from the database. You can schedule this command by using the DEFINE SCHEDULE command. If you schedule the EXPIRE INVENTORY command, set the expiration interval to 0 (zero) in the server options so that the server does not run expiration processing when you start the server.

You can control how long the expiration process runs by using the DURATION parameter with the EXPIRE INVENTORY command.

When expiration processing runs, the server normally sends detailed messages about policy changes made since the last time expiration processing ran. You can reduce those messages by using the EXPQUIET server option, or by using the QUIET=YES parameter with the EXPIRE INVENTORY command.. When you use the quiet option or parameter, the server issues messages about policy changes during expiration processing only when files are deleted, and either the default management class or retention grace period for the domain has been used to expire the files.

Additional Expiration Processing with Tivoli Disaster Recovery Manager

If you have Tivoli Disaster Recovery Manager (DRM), one or more database backup volumes may also be deleted during expiration processing if the following conditions are true:

- The volume has a device type of SERVER
- The volume is not part of the most recent database backup series
- The last volume of the database backup series has exceeded the expiration value specified with the SET DRMDBBACKUPEXPIREDAYS command

See “Moving Backup Volumes Onsite” on page 561 for more information.

Configuring Policy for Specific Cases

This section includes recommendations for some cases for which policy changes may be needed.

- “Configuring Policy for Direct-to-Tape Backups”
- “Configuring Policy for Tivoli Data Protection Application Clients” on page 304
- “Policy for Logical Volume Backups” on page 304
- “Configuring Policy for Managed System for SAN” on page 306
- “Policy for Tivoli Storage Manager Servers as Clients” on page 308
- “Setting Policy to Enable Point-in-Time Restore for Clients” on page 308

Configuring Policy for Direct-to-Tape Backups

The server default policy enables client nodes to back up data to disk storage pools on the server. As an alternative, you may configure a policy to store client data directly in tape storage pools to reduce contention for disk resources. If you back up directly to tape, the number of clients that can back up data at the same time is equal to the number of drives available to the storage pool (through the mount limit of the device class). For example, if you have one drive, only one client at a time can back up data.

The direct-to-tape backup eliminates the need to migrate data from disk to tape. However, performance of tape drives is often lower when backing up directly to tape than when backing up to disk and then migrating to tape. Backing up data directly to tape usually means more starting and stopping of the tape drive. Backing up to disk then migrating to tape usually means the tape drive moves more continuously, meaning better performance.

At the server command line, you may define a new policy domain that enables client nodes to back up or archive data directly to tape storage pools. For example, you may define a policy domain named DIR2TAPE with the following steps:

1. Copy the default policy domain STANDARD as a template:

   ```
   copy domain standard dir2tape
   ```

   This command creates the DIR2TAPE policy domain that contains a default policy set, management class, backup and archive copy group, each named STANDARD.
2. Update the backup or archive copy group in the DIR2TAPE policy domain to specify the destination to be a tape storage pool. For example, to use a tape storage pool named TAPEPOOL for backup, enter the following command:

   ```
   update copygroup dir2tape standard standard destination=tapepool
   ```

   To use a tape storage pool named TAPEPOOL for archive, enter the following command:

   ```
   update copygroup dir2tape standard standard type=archive
    destination=tapepool
   ```
3. Activate the changed policy set.

   ```
   activate policyset dir2tape standard
   ```
4. Assign client nodes to the DIR2TAPE policy domain. For example, to assign a client node named TAPEUSER1 to the DIR2TAPE policy domain, enter the following command:

   ```
   update node tapeuser1 domain=dir2tape
   ```

Configuring Policy for Tivoli Data Protection Application Clients

The Tivoli Data Protection application clients using the server to store data may require that you configure policy to make the most efficient use of server storage. See the user’s guide for each application client for policy requirements.

Some of the application clients include a time stamp in each database backup. Because the default policy for the server keeps one backup version of each unique file, database backups managed by default policy are never deleted because each backup is uniquely named with its time stamp. To ensure that the server deletes backups as required, configure policy as recommended in the user’s guide for the application client.

Policy for Logical Volume Backups

Consider defining a management class specifically for logical volume backups. To enable clients to restore a logical volume and then reconcile the results of any file backup operations since the logical volume backup was made, you must set up management classes with the backup copy group set up differently from the STANDARD. The Versions Data Exists, Versions Data Deleted, and Retain Extra Versions parameters work together to determine over what time period a client can restore a logical volume image and reconcile later file backups. Also, you may have server storage constraints that require you to control the number of backup versions allowed for logical volumes.

Backups of logical volumes are intended to help speed the restoration of a machine. One way to use the capability is to have users periodically (for example, once a month) perform a logical volume backup, and schedule daily full incremental backups. If a user restores a logical volume, the program first restores the logical volume backup and then any files that were changed since the backup (incremental or other file backup processes). The user can also specify that the restore process reconcile any discrepancies that can result when files are deleted.

For example, a user backs up a logical volume, and the following week deletes one or more files from the volume. At the next incremental backup, the server records in its database that the files were deleted from the client. When the user restores the logical volume, the program can recognize that files have been deleted since the backup was created. The program can delete the files as part of the restore process. To ensure that users can use the capability to reconcile later incremental backups with a restored logical volume, you need to ensure that you coordinate policy for incremental backups with policy for backups for logical volumes.

For example, you decide to ensure that clients can choose to restore files and logical volumes from any time in the previous 60 days. You can create two management classes, one for files and one for logical volumes. Table 25 shows the relevant parameters. In the backup copy group of both management classes, set the Retain Extra Versions parameter to 60 days.

In the management class for files, set the parameters so that the server keeps versions based on age rather than how many versions exist. More than one backup version of a file may be stored per day if clients perform selective backups or if clients perform incremental backups more than once a day. The Versions Data Exists parameter and the Versions Data Deleted parameter control how many of these versions are kept by the server. To ensure that any number of backup versions are kept for the required 60 days, set both the Versions Data Exists parameter and the Versions Data Deleted parameter to NOLIMIT for the management class for files. This means that the server retains backup versions based on how old the versions are, instead of how many backup versions of the same file exist.

For logical volume backups, the server ignores the frequency attribute in the backup copy group.

Table 25. Example of Backup Policy for Files and Logical Volumes

Parameter (backup copy group in the management class)	Management Class for Files	Management Class for Logical Volumes
Versions Data Exists	NOLIMIT	3 versions
Versions Data Deleted	NOLIMIT	1
Retain Extra Versions	60 days	60 days
Retain Only Version	120 days	120 days

Configuring Policy for Tivoli Data Protection for NDMP

With the Tivoli Data Protection for NDMP product, you can register a NAS file server as a node. Under the direction of the Tivoli Storage Manager server, the NAS file server performs backup and restore of file system images to a tape library. The TSM server initiates the backup, allocates a drive, and selects and mounts the media. The NAS file server then transfers the data to tape.

Because the NAS file server performs the backup, the data is stored in its own format. For Network Appliance file servers, the data is stored in the NETAPPDUMP data format. To manage NAS file server image backups, copy groups for NAS nodes must point to a storage pool that has a data format of NETAPPDUMP. To set up the required policy for NAS nodes, you can define a new, separate policy domain. See “Chapter 6. Setting Up Tivoli Data Protection for NDMP” on page 97 for details. The following backup copy group attributes are ignored for NAS images:

- Frequency
- Mode
- Retain Only Versions
- Serialization
- Versions Data Deleted

Backups for NAS nodes can be initiated from the server, or from a client that has authority over the NAS node. For client-initiated backups, you can use client option sets that contain include and exclude statements to bind NAS file system images to a specific management class. The valid options that can be used for a NAS node are: include.fs.nas, exclude.fs.nas, and domain.nas. For details on the options see the *Backup-Archive Installation and User’s Guide* for your particular client platform. For more information about client option sets see “Creating Client Option Sets from the Server” on page 253.

Configuring Policy for Managed System for SAN

With the Managed System for SAN feature, you can set up a SAN configuration in which a client directly accesses a storage device to read or write data. SAN data transfer requires setup on the server and on the client, and the installation of a storage agent on the client machine. The storage agent transfers data between the client and the storage device. See *TSM Managed System for SAN Storage Agent User’s Guide* for details. See the Web site for details on clients that support the feature: http://www.tivoli.com/support/storage_mgr/tivolimain.html.

One task in configuring your systems to use this feature is to set up policy for the clients. Copy groups for these clients must point to the storage pool that is associated with the SAN devices. (“Configuring Tivoli Storage Manager for LAN-free Data Movement” on page 92 describes how to configure the devices and define the storage pool.) Clients for which you have mapped the SAN drives in this storage pool can then use the SAN to send data directly to the device for backup, archive, restore, and retrieve.

To set up the required policy, either define a new, separate policy domain, or define a new management class in an existing policy domain:

- “Define a New Policy Domain” on page 307
- “Define a New Management Class in an Existing Policy Domain” on page 308

Define a New Policy Domain

One way to configure policy for clients is to define a separate policy domain in which the active policy set has a default management class with the required settings. Then register all clients using SAN data transfer to that domain. Do the following:

1. Create the policy domain for the clients. For example, to define a policy domain that is named SANCLIENTS, enter the following command:

   ```
   define domain sanclients
    description='Policy domain for clients using SAN devices'
   ```

2. Create a policy set in that domain. For example, to define the policy set that is named BASE in the SANCLIENTS policy domain, enter the following command:

   ```
   define policyset sanclients base
   ```

3. Create the default management class for the policy set. First define the management class, then assign the management class as the default for the policy set.

   For example, to define the management class that is named SANCLIENTMC, enter the following command:

   ```
   define mgmtclass sanclients base sanclientmc
   ```

   Then assign the new management class as the default:

   ```
   assign defmgmtclass sanclients base sanclientmc
   ```

4. Define the backup copy group in the default management class, as follows:

   - Specify the DESTINATION, the name of the storage pool that is associated with the SAN devices on the server.

     The storage pool must already be set up. The storage pool must point to a device class that is associated with the library for the SAN devices. See "Configuring Tivoli Storage Manager for LAN-free Data Movement" on page 92 for details.
   - Accept the default settings for all remaining parameters.

   For example, to define the backup copy group for the SANCLIENTMC management class, enter the following command:

   ```
   define copygroup sanclients base sanclientmc standard destination=sanpool
   ```

5. Define the archive copy group in the default management class, as follows:

   - Specify the DESTINATION, the name of the storage pool that is associated with the SAN devices on the server.

     The storage pool must already be set up. The storage pool must point to a device class that is associated with the library for the SAN devices. See "Configuring Tivoli Storage Manager for LAN-free Data Movement" on page 92 for details.
   - Accept the default settings for all remaining parameters.

   For example, to define the archive copy group for the SANCLIENTMC management class, enter the following command:

   ```
   define copygroup sanclients base sanclientmc standard
    type=archive destination=sanpool
   ```

6. Activate the policy set.

   For example, to activate the BASE policy set in the SANCLIENTS policy domain, enter the following command:

   ```
   activate policyset sanclients base
   ```

7. Register or update the application clients to associate them with the new policy domain.

   For example, to update the node SANCLIENT1, enter the following command:

   ```
   update node sanclient1 domain=sanclients
   ```

Define a New Management Class in an Existing Policy Domain

If you choose not to define a separate policy domain with the appropriate management class as the default, you must define a new management class within an existing policy domain and activate the policy set. Because the new management class is not the default for the policy domain, you must add an include statement to each client options file to bind objects to that management class.

For example, suppose `sanclientmc` is the name of the management class that you defined for clients that are using devices on a SAN. You want the client to be able to use the SAN for backing up any file on the *c* drive. Put the following line at the end of the client's include-exclude list:

```
include c:* sanclientmc
```

For details on the include-exclude list, see *Backup-Archive Installation and User's Guide*.

Policy for Tivoli Storage Manager Servers as Clients

One server (a source server) can be registered as a client to another server (the target server). Data stored by the source server appears as archived files on the target server. The source server is registered to a policy domain on the target server, and uses the default management class for that policy domain. In the default management class, the destination for the archive copy group determines where the target server stores data for the source server. Other policy specifications, such as how long to retain the data, do not apply to data stored for a source server. See "Using Virtual Volumes to Store Data on Another Server" on page 469 for more information.

Setting Policy to Enable Point-in-Time Restore for Clients

To enable clients to restore backed-up files to a specific point in time, you must set up the backup copy group differently from the STANDARD. The Versions Data Exists, Versions Data Deleted, and Retain Extra Versions parameters work together to determine over what time period a client can perform a point-in-time restore operation.

For example, you decide to ensure that clients can choose to restore files from anytime in the previous 60 days. In the backup copy group, set the Retain Extra Versions parameter to 60 days. More than one backup version of a file may be stored per day if clients perform selective backups or if clients perform incremental backups more than once a day. The Versions Data Exists parameter and the Versions Data Deleted parameter control how many of these versions are kept by the server. To ensure that any number of backup versions are kept for the required 60 days, set both the Versions Data Exists parameter and the Versions Data Deleted parameter to NOLIMIT. This means that the server essentially determines the backup versions to keep based on how old the versions are, instead of how many backup versions of the same file exist.

Keeping backed-up versions of files long enough to allow clients to restore their data to a point in time can mean increased resource costs. Requirements for server storage increase because more file versions are kept, and the size of the server database increases to track all of the file versions. Because of these increased costs, you may want to choose carefully which clients can use the policy that allows for point-in-time restore operations.

Clients need to run full incremental backup operations frequently enough so that Tivoli Storage Manager can detect files that have been deleted on the client file system. Only a full incremental backup can detect whether files have been deleted since the last backup. If full incremental backup is not done often enough, clients who restore to a specific time may find that many files that had actually been deleted from the workstation get restored. As a result, a client's file system may run out of space during a restore process.

Distributing Policy Using Enterprise Configuration

If you set up one TSM server as a configuration manager, you can distribute policy to other TSM servers. To distribute policy, you associate a policy domain with a profile. Managed servers that subscribe to the profile then receive the following definitions:

- The policy domain itself
- Policy sets in that domain, except for the ACTIVE policy set
- Management classes in the policy sets
- Backup and archive copy groups in the management classes
- Client schedules associated with the policy domain

The names of client nodes and client-schedule associations are not distributed. The ACTIVE policy set is also not distributed.

The distributed policy becomes managed objects (policy domain, policy sets, management classes, and so on) defined in the database of each managed server. To use the managed policy, you must activate a policy set on each managed server. If storage pools specified as destinations in the policy do not exist on the managed server, you receive messages pointing out the problem when you activate the policy set. You can create new storage pools to match the names in the policy set, or you can rename existing storage pools.

On the managed server you also must associate client nodes with the managed policy domain and associate client nodes with schedules.

See "Setting Up an Enterprise Configuration" on page 443 for details.

Querying Policy

Task	Required Privilege Class
Query any policy domain, policy set, management class, or copy group	Any administrator

You can request information about the contents of policy objects. You might want to do this before creating new objects or when helping users to choose policies that fit their needs.

You can specify the output of a query in either standard or detailed format. The examples in this section are in standard format.

On a managed server, you can see whether the definitions are managed objects. Request the detailed format in the query and check the contents of the `Last update by (administrator)` field. For managed objects, this field contains the string $$CONFIG_MANAGER$$.

Querying Copy Groups

To request information about backup copy groups (the default) in the ENGPOLDOM engineering policy domain, enter:

```
query copygroup engpoldom * *
```

The following shows the output from the query. It shows that the ACTIVE policy set contains two backup copy groups that belong to the MCENG and STANDARD management classes.

```
Policy     Policy     Mgmt       Copy      Versions  Versions    Retain   Retain
Domain     Set Name   Class      Group         Data      Data     Extra     Only
Name                  Name       Name        Exists   Deleted  Versions  Version
---------  ---------  ---------  --------  --------  --------  --------  -------
ENGPOLDOM  ACTIVE     MCENG      STANDARD         5         4        90      600
ENGPOLDOM  ACTIVE     STANDARD   STANDARD         2         1        30       60
ENGPOLDOM  STANDARD   MCENG      STANDARD         5         4        90      600
ENGPOLDOM  STANDARD   STANDARD   STANDARD         2         1        30       60
ENGPOLDOM  TEST       STANDARD   STANDARD         2         1        30       60
```

To request information about archive copy groups in the ENGPOLDOM engineering policy domain, enter:

```
query copygroup engpoldom * type=archive
```

The following shows the output from the query.

```
Policy        Policy        Mgmt          Copy             Retain
Domain        Set Name      Class         Group           Version
Name                        Name          Name
---------     ---------     ---------     ---------     --------
ENGPOLDOM     ACTIVE        MCENG         STANDARD           730
ENGPOLDOM     ACTIVE        STANDARD      STANDARD           365
ENGPOLDOM     STANDARD      MCENG         STANDARD           730
ENGPOLDOM     STANDARD      STANDARD      STANDARD           365
ENGPOLDOM     TEST          STANDARD      STANDARD           365
```

Querying Management Classes

To request information about management classes in the ENGPOLDOM engineering policy domain, enter:

```
query mgmtclass engpoldom * *
```

The following figure is the output from the query. It shows that the ACTIVE policy set contains the MCENG and STANDARD management classes.

```
Policy       Policy       Mgmt         Default      Description
Domain       Set Name     Class        Mgmt
Name                      Name         Class ?
---------    ---------    ---------    ---------    ------------------------
ENGPOLDOM    ACTIVE       MCENG        No          Engineering Management
                                                    Class with Backup and
                                                    Archive Copy Groups
ENGPOLDOM    ACTIVE       STANDARD     Yes         Installed default
                                                    management class
ENGPOLDOM    STANDARD     MCENG        No          Engineering Management
                                                    Class with Backup and
                                                    Archive Copy Groups
ENGPOLDOM    STANDARD     STANDARD     Yes         Installed default
                                                    management class
ENGPOLDOM    TEST         STANDARD     Yes         Installed default
                                                    management class
```

Querying Policy Sets

To request information about policy sets in the ENGPOLDOM engineering policy domain, enter:

```
query policyset engpoldom *
```

The following figure is the output from the query. It shows an ACTIVE policy set and two inactive policy sets, STANDARD and TEST.

```
Policy        Policy        Default       Description
Domain        Set Name      Mgmt
Name                        Class
                            Name
---------     ---------     ---------     ------------------------
ENGPOLDOM     ACTIVE        STANDARD      Installed default policy
                                           set
ENGPOLDOM     STANDARD      STANDARD      Installed default policy
                                           set
ENGPOLDOM     TEST          STANDARD      Policy set for testing
```

Querying Policy Domains

To request information about a policy domain (for example, to determine if any client nodes are registered to that policy domain), enter:

```
query domain *
```

The following figure is the output from the query. It shows that both the ENGPOLDOM and STANDARD policy domains have client nodes assigned to them.

```
Policy       Activated    Activated     Number of    Description
Domain       Policy       Default      Registered
Name         Set          Mgmt              Nodes
                          Class
---------    ---------    ---------    ----------    ------------------------
APPCLIEN-    BASE         APPCLIEN-             1    Policy domain for
 TS                        TMC                        application clients
ENGPOLDOM    STANDARD     STANDARD             21    Engineering Policy
                                                      Domain
STANDARD     STANDARD     STANDARD             18    Installed default policy
                                                      domain.
```

Deleting Policy

When you delete a policy object, you also delete any objects belonging to it. For example, when you delete a management class, you also delete the copy groups in it.

You cannot delete the ACTIVE policy set or objects that are part of that policy set.

Task	Required Privilege Class
Delete policy domains	System
Delete any policy sets, management classes, or copy groups	System or unrestricted policy
Delete policy sets, management classes, or copy groups that belong to policy domains over which you have authority	Restricted policy

You can delete the policy objects named STANDARD that come with the server. However, all STANDARD policy objects are restored whenever you reinstall the server. If you reinstall the server after you delete the STANDARD policy objects, the server issues messages during processing of a subsequent DSMSERV AUDITDB command. The messages may include the following statement: “An instance count does not agree with actual data.” The DSMSERV AUDITDB command corrects this problem by restoring the STANDARD policy objects. If necessary, you can later delete the restored STANDARD policy objects.

Deleting Copy Groups

You can delete a backup or archive copy group if it does not belong to a management class in the ACTIVE policy set.

For example, to delete the backup and archive copy groups belonging to the MCENG and STANDARD management classes in the STANDARD policy set, enter:

```
delete copygroup engpoldom standard mceng type=backup

delete copygroup engpoldom standard standard type=backup

delete copygroup engpoldom standard mceng type=archive

delete copygroup engpoldom standard standard type=archive
```

Deleting Management Classes

You can delete a management class if it does not belong to the ACTIVE policy set.

For example, to delete the MCENG and STANDARD management classes from the STANDARD policy set, enter:

```
delete mgmtclass engpoldom standard mceng
delete mgmtclass engpoldom standard standard
```

When you delete a management class from a policy set, the server deletes the management class and all copy groups that belong to the management class in the specified policy domain.

Deleting Policy Sets

Authorized administrators can delete any policy set other than the ACTIVE policy set. For example, to delete the TEST policy set from the ENGPOLDOM policy domain, enter:

```
delete policyset engpoldom test
```

When you delete a policy set, the server deletes all management classes and copy groups that belong to the policy set within the specified policy domain.

The ACTIVE policy set in a policy domain cannot be deleted. You can replace the contents of the ACTIVE policy set by activating a different policy set. Otherwise, the only way to remove the ACTIVE policy set is to delete the policy domain that contains the policy set.

Deleting Policy Domains

You can delete a policy domain only if the domain has no client nodes registered to it. To determine if any client nodes are registered to a policy domain, issue the QUERY DOMAIN or the QUERY NODE command. Move any client nodes to another policy domain, or delete the nodes.

For example, to delete the STANDARD policy domain, perform the following steps:

1. Request a list of all client nodes assigned to the STANDARD policy domain by entering:
   ```
   query node * domain=standard
   ```
2. If client nodes are assigned to the policy domain, remove them in one of the following ways:
   - Assign each client node to a new policy domain. For example, enter the following commands:
     ```
     update node htang domain=engpoldom
     update node tomc domain=engpoldom
     update node pease domain=engpoldom
     ```

     If the ACTIVE policy set in ENGPOLDOM does not have the same management class names as in the ACTIVE policy set of the STANDARD policy domain, then backup versions of files may be bound to a different management class name, as described in “How Files and Directories Are Associated with a Management Class” on page 282.
   - Delete each node from the STANDARD policy domain by first deleting all file spaces belonging to the nodes, then deleting the nodes.
3. Delete the policy domain by entering:
   ```
   delete domain standard
   ```

When you delete a policy domain, the server deletes the policy domain and all policy sets (including the ACTIVE policy set), management classes, and copy groups that belong to the policy domain.

Chapter 13. Managing Data for Client Nodes

This chapter contains information to help you generate backup sets and enable subfile backups for client nodes. You can also use data validation for client nodes so that any data corruption is identified when data is sent over the network between the client and server.

See the following sections for more information:

Tasks:
"Validating a Node's Data During a Client Session" on page 316
"Generating Client Backup Sets on the Server" on page 317
"Restoring Backup Sets from a Backup-Archive Client" on page 318
"Moving Backup Sets to Other Servers" on page 319
"Managing Client Backup Sets" on page 319
"Enabling Clients to Use Subfile Backup" on page 322
Concepts:
"Choosing Where to Enable Data Validation" on page 532
"Performance Considerations" on page 316
"Creating and Using Client Backup Sets" on page 316

Validating a Node's Data

Data validation can identify data corruption during a client session when data is sent between a client and the server. Tivoli Storage Manager provides the option of specifying whether a cyclic redundancy check (CRC) is performed during a client session to validate data sent over the network between a client or a storage agent and the server.

Cyclic redundancy checking is performed at the client when the client requests services from the server. For example, the client issues a query, backup, or archive request. The server also performs a CRC operation on the data sent by the client and compares its value with the value calculated by the client. If the CRC values do not match, the server will issue an error message once per session. Depending on the operation, the client may attempt to automatically retry the operation.

After Tivoli Storage Manager completes the data validation, the client and server discard the CRC values generated in the current session.

Data validation can be enabled for one or all of the following:

- Tivoli Storage Manager client nodes at Version 5.1.
- Tivoli Storage Manager storage agents at Version 5.1. For details refer to the storage agent user's guide for your particular operating system.

This section provides information about data validation between a node and the server.

© Copyright IBM Corp. 1993, 2002

See “Choosing Where to Enable Data Validation” on page 532 to help you determine where to enable data validation.

Performance Considerations

Consider the impact on performance when you decide whether data validation is necessary for all nodes or some nodes. Data validation impacts performance because additional CPU overhead is required on both the client and server to calculate and compare CRC values.

This type of validation is independent from validating data written to a storage pool volume. See “Data Validation During Audit Volume Processing” on page 531.

Validating a Node’s Data During a Client Session

You can enable data validation for a node by using the REGISTER NODE or UPDATE NODE command. By default, data validation is set to NO.

Methods for enabling data validation for a node include choosing data validation for individual nodes, specifying a set of nodes by using a wildcard search string, or specifying a group of nodes in a policy domain.

For example, to enable data validation for existing node, ED, you can issue an UPDATE NODE command. This user backs up the company payroll records weekly and you have decided it is necessary to have all the user data validated: the data itself and metadata.

```
update node ed validateprotocol=all
```

Later, the network has shown to be stable and no data corruption has been identified when user ED has processed backups. You can then disable data validation to minimize the performance impact of validating all of ED’s data during a client session. For example:

```
update node ed validateprotocol=no
```

Creating and Using Client Backup Sets

A *backup set* is a collection of backed-up data from one client, stored and managed as a single object on specific media in server storage. The server creates copies of active versions of a client’s backed up objects that are within the one or more file spaces specified with the GENERATE BACKUPSET command, and consolidates them onto sequential media. Currently, the backup object types supported for backup sets include directories and files only. The process is also called *instant archive*.

The media may be directly readable by a device such as a CD-ROM, JAZ, or ZIP drive attached to a client’s machine.

Administrators can generate multiple copies of backup sets that correspond to some point-in-time. The backup sets can be retained for various time periods. This is an efficient way to create long-term storage of periodic backups, without requiring the data to be sent over the network again.

While an administrator can generate a backup set from any client’s backed up files, backup sets can only be used by a backup-archive client. You cannot generate a backup set for a NAS node.

See the following sections for details:

- “Generating Client Backup Sets on the Server”
- “Restoring Backup Sets from a Backup-Archive Client” on page 318
- “Moving Backup Sets to Other Servers” on page 319
- “Managing Client Backup Sets” on page 319

Generating Client Backup Sets on the Server

Task	Required Privilege Class
Generate a backup set	System or restricted policy over the domain to which the node is assigned

You can generate backup sets on the server for client nodes. The client node for which a backup set is generated must be registered to the server. An incremental backup must be completed for a client node before the server can generate a backup set for the client node.

The GENERATE BACKUPSET command runs as a background process on the server. If you cancel the background process created by this command, the media may not contain a complete backup set.

See the following sections:

- “Choosing Media for Generating the Backup Set”
- “Selecting a Name for the Backup Set” on page 318
- “Setting a Retention Period for the Backup Set” on page 318
- “Example: Generating a Client Backup Set” on page 318

Choosing Media for Generating the Backup Set

To generate a backup set, you must specify a device class that is associated with the media to which the backup set will be written.

Consider the following when you select a device class for writing the backup set:

- Generate the backup set on any sequential access devices whose device types are supported on **both** the client and server machines. If you do not have access to compatible devices, you will need to define a device class for a device type that is supported on both the client and server.
- Ensure that the media type and recording format used for generating the backup set is supported by the device that will be reading the backup set.

You can write backup sets to sequential media: sequential tape and device class FILE. The tape volumes containing the backup set are not associated with storage pools and, therefore, are not migrated through the storage pool hierarchy.

For device class FILE, the server creates each backup set with a file extension of OST. You can copy FILE device class volumes to removable media that is associated with CD-ROM, JAZ, or ZIP devices, by using the REMOVABLEFILE device type. For more information, see “Configuring Removable File Devices” on page 86.

Using Scratch Media: You can determine whether to use scratch volumes when you generate a backup set. If you do not use specific volumes, the server uses scratch volumes for the backup set.

You can use specific volumes for the backup set. If there is not enough space to store the backup set on the volumes, the server uses scratch volumes to store the remainder of the backup set.

Selecting a Name for the Backup Set

The server adds a unique suffix to the name you specify for the backup set. For example, if you name the backup set *mybackupset*, the server adds a unique extension, such as 3099, to the name. This allows you to create backup sets with the same name without overwriting previous backup sets.

To later display information about this backup set, you can include a wildcard character with the name, such as *mybackupset**, or you can specify the fully qualified name, such as *mybackupset.3099*.

Setting a Retention Period for the Backup Set

You can set the retention period, specified as a number of days, to retain the backup set on the server. You can specify a number between zero and 9999 days. Backup sets are retained on the server for 365 days if you do not specify a value. The server uses the retention period to determine when to expire the volumes on which the backup set resides.

Example: Generating a Client Backup Set

Generate a backup set on portable media that the client can later use to restore the data. Use the following steps to generate a backup set on a CD-ROM:

1. Define a library whose type is MANUAL. Name the library MANUALLIB.
   ```
   define library manuallib libtype=manual
   ```
2. Define a device class whose device type is REMOVABLEFILE. Name the device class BACKSET:
   ```
   define devclass backset devtype=removablefile library=manuallib
   ```
3. Define a drive to associate with the library. Name the drive CDDRIVE and the device /cdrom
   ```
   define drive manuallib cddrive device=/cdrom
   ```
4. Define a device class whose device type is FILE. Name the device class FILES:
   ```
   define devclass files devtype=file maxcapacity=640M dir=/backupset
   ```
5. Generate the backup set to the FILE device class for client node JOHNSON. Name the backup set PROJECT and retain it for 90 days.
   ```
   generate backupset johnson project devclass=file scratch=yes
   retention=90
   ```
6. Use your own software for writing CD-ROMs. For this example, the CD-ROM volume names are VOL1, VOL2, and VOL3. These names were put on the CD-ROM as they were created.

   For an example of using the backup set on the CD-ROM, see “Moving Backup Sets to Other Servers” on page 319.

Restoring Backup Sets from a Backup-Archive Client

Backup-archive client nodes can restore their backup sets in either of two ways:

- Directly from the server.
- Using a device attached to the client’s machine that will read the media in which the backup set is stored.

Backup sets can only be used by a backup-archive client, and only if the files in the backup set originated from a backup-archive client.

For more information about restoring backup sets, see *Using the Backup-Archive Client* guide for your particular operating system.

Moving Backup Sets to Other Servers

Task	Required Privilege Class
Define a backup set	If the REQSYSAUTHOUTFILE server option is set to YES, system privilege is required. If the REQSYSAUTHOUTFILE server option is set to NO, system or restricted policy over the domain to which the node is assigned is required.

You can define (move) a backup set generated on one server to another Tivoli Storage Manager server. Any client backup set that you generate on one server can be defined to another server as long as the servers share a common device type. The level of the server defining the backup set must be equal to or greater than the level of the server that generated the backup set.

If you have multiple servers connecting to different clients, the DEFINE BACKUPSET command makes it possible for you to take a previously generated backup set and make it available to other servers. The purpose is to allow the user flexibility in moving backup sets to different servers, thus allowing the user the ability to restore their data from a server other than the one on which the backup set was created.

Using the example described in "Example: Generating a Client Backup Set" on page 318, you can make the backup set that was copied to the CD-ROM available to another server by entering:

```
define backupset johnson project devclass=cdrom volumes=vol1,vol2,vol3
description="backup set copied to a CD-ROM"
```

Managing Client Backup Sets

You can update, query, and delete backup sets.

Task	Required Privilege Class
Update the retention period assigned to a backup set	System or restricted policy over the domain to which the node is assigned
Display information about backup sets	Any administrator
Display information about backup set contents	System or restricted policy over the domain to which the node is assigned
Delete backup set	If the REQSYSAUTHOUTFILE server option is set to YES, system privilege is required. If the REQSYSAUTHOUTFILE server option is set to NO, system or restricted policy over the domain to which the node is assigned is required.

Updating the Retention Period of a Backup Set

When you want to change the number of days the server retains a backup set, update the retention period that is associated with the backup set. For example, to update the retention period assigned to backup set named ENGDATA.3099, belonging to client node JANE, to 120 days, enter:

```
update backupset jane engdata.3099 retention=120
```

Displaying Backup Set Information

To view information about backup sets, you can use the QUERY BACKUPSET command. The output that is displayed lists information such as the name of the client node whose data is contained in the backup set as well as the description of the backup set, assuming one has been used.

The following figure shows the report that is displayed after you enter:

```
query backupset
```

```
       Node Name: JANE
  Backup Set Name: MYBACKUPSET.3099
       Date/Time: 06/09/1999 16:17:47
 Retention Period: 60
Device Class Name: DCFILE
      Description:
```

Displaying Information about Backup Set Volumes

A client's backup set can reside on more than one volume. The server records the information about the volumes used for the backup set in the volume history file. Volume history includes information such as the date and time the backup set was generated, the device class to which the backup set was written, and the command used to generate the backup set. If a backup set spans several volumes, the server displays the command used to generate the backup set only with the first volume.

You can view this information when you use the QUERY VOLHISTORY command, with BACKUPSET specified as the volume type.

The following example shows how this particular client's backup set resides on three volumes, and the command used to generate the backup set is displayed with the first volume.

```
      Date/Time: 09/24/2001 07:34:06 PM
    Volume Type: BACKUPSET
  Backup Series:
Backup Operation:
     Volume Seq: 1
   Device Class: FILE
    Volume Name: 01334846.ost
 Volume Location:
        Command: gen backupset client57 testbs /home dev=file scratch=yes
                  ret=2 desc="Client57 backupset"

      Date/Time: 09/24/2001 07:34:06 PM
    Volume Type: BACKUPSET
  Backup Series:
Backup Operation:
     Volume Seq: 2
   Device Class: FILE
    Volume Name: 01334849.ost
 Volume Location:
        Command:


      Date/Time: 09/24/2001 07:34:06 PM
    Volume Type: BACKUPSET
  Backup Series:
Backup Operation:
     Volume Seq: 3
   Device Class: FILE
    Volume Name: 01334850.ost
 Volume Location:
        Command:
```

Displaying Contents of Backup Sets

You can display information about the contents of backup sets by using the QUERY BACKUPSETCONTENTS command. When you issue the query, the server displays only one backup set at a time.

The server displays information about the files and directories that are contained in a backup set. The following figure shows the report that is displayed after you enter:

```
query backupsetcontents jane engdata.3099
```

```
Node Name                Filespace  Client's Name for File
                         Name
------------------------ ---------- -----------------------------------
JANE                     /srvr      /deblock
JANE                     /srvr      /deblock.c
JANE                     /srvr      /dsmerror.log
JANE                     /srvr      /dsmxxxxx.log
JANE                     ...        ......
```

How File Space and File Names May be Displayed: File space names and file names that can be in a different code page or locale than the server do not display correctly on the administrator’s Web interface or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (??), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

Deleting Backup Sets

When the server creates a backup set, the retention period assigned to the backup set determines how long the backup set remains in the database. When that date passes, the server automatically deletes the backup set when expiration processing runs. However, you can also manually delete the client's backup set from the server before it is scheduled to expire by using the DELETE BACKUPSET command.

After a backup set is deleted, the volumes return to scratch status if Tivoli Storage Manager acquired them as scratch volumes. Scratch volumes associated with a device type of FILE are deleted.

To delete a backup set named ENGDATA.3099, belonging to client node JANE, created before 11:59 p.m. on March 18, 1999, enter:

```
delete backupset jane engdata.3099 begindate=03/18/1999 begintime=23:59
```

To delete all backup sets belonging to client node JANE, created before 11:59 p.m. on March 18, 1999, enter:

```
delete backupset jane * begindate=03/18/1999 begintime=23:59
```

Enabling Clients to Use Subfile Backup

A basic problem that remote and mobile users face today is connecting to storage management services by using modems with limited bandwidth or poor line quality. This creates a need for users to minimize the amount of data they send over the network, as well as the time that they are connected to the network.

To help address this problem, you can use subfile backups. When a client's file has been previously backed up, any subsequent backups are *typically* made of the portion of the client's file that has changed *(a subfile)*, rather than the entire file. A *base* file is represented by a backup of the entire file and is the file on which subfiles are dependent. If the changes to a file are extensive, a user can request a backup on the entire file. A new base file is established on which subsequent subfile backups are dependent.

This type of backup makes it possible for mobile users to reduce connection time, network traffic, and the time it takes to do a backup. To enable this type of backup, see "Setting Up Clients to Use Subfile Backup" on page 323.

Example of Subfile Backups

Assume that on a Monday, a user requests an incremental backup of a file called *CUST.TXT*. The user makes daily updates to the CUST.TXT file and requests subsequent backups.

The following table describes how Tivoli Storage Manager handles backups of file CUST.TXT.

Version	Day of subsequent backup	What Tivoli Storage Manager backs up
One	Monday	The entire CUST.TXT file (the base file)
Two	Tuesday	A subfile of CUST.TXT. The server compares the file backed up on Monday with the file that needs to be backed up on Tuesday. A subfile containing the changes between the two files is sent to the server for the backup.
Three	Wednesday	A subfile of CUST.TXT. Tivoli Storage Manager compares the file backed up on Monday with the file that needs to be backed up on Wednesday. A subfile containing the changes between the two files is sent to the server for the backup.

Setting Up Clients to Use Subfile Backup

To enable subfile backup, do the following:

- ***On the server:*** You must set up the server to allow clients to back up subfiles. Use the SET SUBFILE command.

  ```
  set subfile client
  ```

- ***On the clients:*** The SUBFILEBACKUP, SUBFILECACHEPATH, and SUBFILECACHESIZE options must be set in the client's options file (dsm.opt).

  You can control these options from the server by including them in client option sets. For example, you can disable subfile backup for individual client nodes by setting SUBFILEBACKUP=NO in the client option set associated with the client node. See "Creating Client Option Sets from the Server" on page 253 for how to set up and use client option sets.

  See *Tivoli Storage Manager for Windows: Backup-Archive Installation and User's Guide* for more information about the options.

Managing Subfile Backups

The following sections describe how Tivoli Storage Manager manages subfiles that are restored, exported, imported, or added to a backup set.

Restoring Subfiles

When a client issues a request to restore subfiles, Tivoli Storage Manager restores subfiles along with the corresponding base file back to the client. This process is transparent to the client. That is, the client does not have to determine whether all subfiles and corresponding base file were restored during the restore operation.

You can define (move) a backup set that contains subfiles to an earlier version of a server that is not enabled for subfile backup. That server can restore the backup set containing the subfiles to a client not able to restore subfiles. However, this process is not recommended as it could result in a data integrity problem.

Exporting and Importing Subfiles

When subfiles are exported during an export operation, Tivoli Storage Manager also exports the corresponding base file to volumes you specify. When the base file and its dependent subfiles are imported from the volumes to a target server and import processing is canceled while the base file and subfiles are being imported, the server automatically deletes any incomplete base files and subfiles that were stored on the target server.

Expiration Processing of Base Files and Subfiles

Because subfiles are useless without the corresponding base file, the server processes base files eligible for expiration differently. For example, when expiration processing runs, Tivoli Storage Manager recognizes a base file as eligible for expiration but does not delete the file until all its dependent subfiles have expired. For more information on how the server manages file expiration, see “Running Expiration Processing to Delete Expired Files” on page 302.

Adding Subfiles to Backup Sets

When a subfile is added to a backup set, Tivoli Storage Manager includes its corresponding base file with the backup set. If the base file and dependent subfiles are stored on separate volumes when a backup set is created, additional volume mounts may be required to create the backup set.

Deleting Base Files

If a base file is deleted as a result of processing a DELETE VOLUME command, the server recognizes its dependent subfiles and deletes them from the server as well. Subfiles without the corresponding base file are incomplete and useless to the user.

Chapter 14. Scheduling Operations for Client Nodes

This chapter contains information about scheduling the following operations:

- Backing up and restoring client data and Tivoli Data Protection application client data.
- Archiving and retrieving client data.
- Running operating system commands.
- Running macro or command files that contain operating system commands, Tivoli Storage Manager commands, or both. You can schedule a command file to run on clients or application clients.

If you are planning to use Tivoli Decision Support products with Tivoli Storage Manager, you can define a schedule that will run the Decision Support Loader automatically. For information, see "Scheduling the Decision Support Loader with Tivoli Storage Manager" on page 426. For information about the Tivoli Decision Support Guide products, see this URL:
http://www.tivoli.com/support/storage_mgr/tivolimain.html

The following concepts are described in this chapter:

Concepts:
"Prerequisites to Scheduling Operations" on page 326
"Comparing Tivoli Storage Manager Scheduling Across Operating Systems and Components" on page 330
"Commands for Scheduling Client Operations" on page 330

Administrators perform the following activities to schedule Tivoli Storage Manager client operations:

Tasks:
"Scheduling a Client Operation" on page 326 (task overview)
"Defining Client Schedules" on page 326
"Associating Client Nodes with Schedules" on page 327
"Starting the Scheduler on the Clients" on page 327
"Displaying Schedule Information" on page 328
"Creating Schedules for Running Command Files" on page 329
"Updating the Client Options File to Automatically Generate a New Password" on page 329

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

© Copyright IBM Corp. 1993, 2002

Prerequisites to Scheduling Operations

To interact with Tivoli Storage Manager for scheduling operations, a client machine must meet the following prerequisites:

- The client node must be registered with the server. For information, see "Chapter 10. Adding Client Nodes" on page 227.
- The client options file (dsm.opt) must contain the network address of the server that the client will contact for services. See "Connecting Nodes with the Server" on page 231 for more information.
- The scheduler must be started on the client machine. See *Backup-Archive Installation and User's Guide* for details.

Scheduling a Client Operation

To automate client operations, you can define new schedules. To later modify, copy, and delete these schedules, see "Chapter 15. Managing Schedules for Client Nodes" on page 333.

When you define a schedule, you assign it to a specific policy domain. You can define more than one schedule for each policy domain.

To set up a client schedule on the server, perform these steps:

1. Define a schedule (DEFINE SCHEDULE command). ("Defining Client Schedules" on page 326)
2. Associate client nodes with the schedule (DEFINE ASSOCIATION command). ("Associating Client Nodes with Schedules" on page 327)
3. Ensure that the clients start the client scheduler. ("Starting the Scheduler on the Clients" on page 327)
4. Display the schedule information and check that the schedule completed successfully (QUERY SCHEDULE and QUERY EVENT commands). ("Displaying Schedule Information" on page 328)

The following sections describe how to automate a basic client operation, incremental backup.

Defining Client Schedules

Task	Required Privilege Class
Define client schedules for any policy domain	System or unrestricted policy
Define client schedules for specific policy domains	System, unrestricted policy, or restricted policy for those domains

Key information to have when scheduling operations are:

- The operation that needs to run
- The time and day when the operation needs to run
- How often the operation needs to repeat

To define a schedule for daily incremental backups, use the DEFINE SCHEDULE command. You must specify the policy domain to which the schedule belongs and the name of the schedule (the policy domain must already be defined). For example:

```
define schedule engpoldom daily_backup starttime=21:00
duration=2 durunits=hours
```

This command results in the following:

- Schedule *DAILY_BACKUP* is defined for policy domain *ENGPOLDOM*.
- The scheduled action is an incremental backup; this is the default.
- The priority for the operation is 5; this is the default. If schedules conflict, the schedule with the highest priority (lowest number) runs first.
- The schedule window begins at 9:00 p.m., and the schedule itself has 2 hours to start.
- The start window is scheduled every day; this is the default.
- The schedule never expires; this is the default.

To change the defaults, see the DEFINE SCHEDULE command in the *Administrator's Reference*.

Associating Client Nodes with Schedules

Task	Required Privilege Class
Associate client nodes with schedules	System, unrestricted policy, or restricted policy for the policy domain to which the schedule belongs

Client nodes process operations according to the schedules associated with the nodes. To associate client nodes with a schedule, use the DEFINE ASSOCIATION command. A client node can be associated with more than one schedule. However, a node must be assigned to the policy domain to which a schedule belongs.

After a client schedule is defined, you can associate client nodes with it by identifying the following information:

- Policy domain to which the schedule belongs
- List of client nodes to associate with the schedule

To associate the ENGNODE client node with the WEEKLY_BACKUP schedule, both of which belong to the ENGPOLDOM policy domain, enter:

```
define association engpoldom weekly_backup engnode
```

Starting the Scheduler on the Clients

The client scheduler must be started before work scheduled by the administrator can be initiated. Administrators must ensure that users start the Tivoli Storage Manager Scheduler on the client or application client directory, and that the scheduler is running at the schedule start time. After the client scheduler starts, it continues to run and initiates scheduled events until it is stopped.

The way that users start the Tivoli Storage Manager Scheduler varies, depending on the operating system that the machine is running. The user can choose to start the client scheduler automatically when the operating system is started, or can start it manually at any time. The user can also have the client acceptor manage the scheduler, starting the scheduler only when needed. For instructions on these tasks, see *Backup-Archive Installation and User's Guide* and the appropriate client user's guide.

Note: Tivoli Storage Manager does not recognize changes that you made to the client options file while the scheduler is running. For Tivoli Storage Manager to use the new values immediately, you must stop the scheduler and restart it.

Displaying Schedule Information

Task	Required Privilege Class
Display information about scheduled operations	Any administrator

You can display information about schedules and whether the schedules ran successfully.

Displaying Schedule Details

When you request information about schedules, the server displays the following information:

- Schedule name
- Policy domain name
- Type of operation to perform
- Start date and time for the initial startup window
- Duration of the startup window
- Time period between startup windows
- Day of the week on which scheduled operations can begin

The following output shows an example of a report that is displayed after you enter:

```
query schedule engpoldom
```

```
Domain       * Schedule Name    Action Start Date/Time      Duration Period Day
------------ - ---------------- ------ -------------------- -------- ------ ---
ENGPOLDOM      MONTHLY_BACKUP   Inc Bk 07/21/1998 12:45:14      2 H    2 Mo Sat
ENGPOLDOM      WEEKLY_BACKUP    Inc Bk 07/21/1998 12:46:21      4 H    1 W  Sat
```

Checking the Status of Scheduled Operations

For Tivoli Storage Manager, a schedule completes successfully as long as the command associated with the schedule is successfully issued. The success of the issued command is independent of the success of the schedule.

- To determine the success of a scheduled operation, query the server. Each scheduled client operation is called an *event*, and is tracked by the server. You can get information about projected and actual scheduled processes by using the QUERY EVENT command. You can get information about scheduled processes that did not complete successfully by using exception reporting with this command.

  For example, you can issue the following command to find out which events were missed (did not start) in the ENGPOLDOM policy domain for the WEEKLY_BACKUP schedule in the previous week:

  ```
  query event engpoldom weekly_backup begindate=-7 begintime=now
  enddate=today endtime=now exceptionsonly=yes
  ```

  For more information about managing event records, see "Managing Event Records" on page 336.

- To determine the success of the commands issued as the result of a successful schedule, you can:

– Check the client’s schedule log.

  The schedule log is a file that contains information such as the statistics about the backed-up objects, the name of the server backing up the objects, and the time and date of the next scheduled operation. By default, Tivoli Storage Manager stores the schedule log as a file called *dsmsched.log* and places the file in the directory where the Tivoli Storage Manager backup-archive client is installed. See the client user’s guide for more information.

– Check the server’s activity log.

  Search or query the activity log for related messages. For example, search for messages that mention the client node name, within the time period that the schedule ran. For example:

```
query actlog begindate=02/23/2001 enddate=02/26/2001 originator=client
nodename=hermione
```

Creating Schedules for Running Command Files

For some clients, you may want to run a command for a different application before running a Tivoli Storage Manager backup. For example, you may want to stop a database application, back up files with Tivoli Storage Manager, and then restart the application. To do this, you can schedule the running of a command file. Application clients *require* schedules that run command files.

A command file (also known as a macro or batch file on different operating systems) is stored on the client. This file contains a sequence of commands that are intended to be run during a scheduled start date and time window. Commands can include operating system commands, the Tivoli Storage Manager client’s DSMC command, and commands for other applications.

To use command files, administrators must create schedules with the ACTION=MACRO parameter. For example, you can define a schedule called DAILY_INCR that will process a command file called *c:\incr.cmd* on the client:

```
define schedule standard daily_incr description="daily incremental file"
 action=macro objects="c:\incr.cmd" starttime=18:00 duration=5
 durunits=minutes period=1 perunits=day dayofweek=any
```

Associate the client with the schedule and ensure that the scheduler is started on the client or application client directory. The schedule runs the file called *c:\incr.cmd* once a day between 6:00 p.m. and 6:05 p.m., every day of the week.

Updating the Client Options File to Automatically Generate a New Password

If the server uses password authentication, clients must use passwords. Passwords are then also required for the server to process scheduled operations for client nodes. If the password expires and is not updated, scheduled operations fail. You can prevent failed operations by allowing Tivoli Storage Manager to generate a new password when the current password expires. Include the PASSWORDACCESS GENERATE option in the Tivoli Storage Manager client options file to ensure that the client always gets a new password when the old one expires.

The PASSWORDACCESS GENERATE option is also required in other situations, such as when you want to use the Web backup-archive client to access a client node. See the client user’s guide for more information.

Comparing Tivoli Storage Manager Scheduling Across Operating Systems and Components

The Tivoli Storage Manager Scheduler provides the capability for the server to process scheduled operations. For client nodes running on operating systems other than Windows, the scheduler is installed when the backup-archive client or application client software is installed. The scheduler can be started by using the DSMC SCHEDULE command on the client's machine. For Windows NT and Windows 2000 clients, the Tivoli Storage Manager Scheduler service must be configured separately.

For more information about installing, configuring, and starting the Tivoli Storage Manager Scheduler, see *Backup-Archive Installation and User's Guide*.

The following table compares the scheduling environment across operating systems and components:

Component Type	Operating System	Scheduling Environment
Backup-Archive Client	UNIX, platforms other than Windows	The scheduler is installed as part of the client installation
Backup-Archive Client	Windows	The scheduler is installed but configured separately
Tivoli Data Protection for Oracle (application client)	AIX	The scheduler is installed as part of the client installation
Tivoli Data Protection for Oracle (application client)	HP-UX	The API must be installed
Tivoli Data Protection for Oracle Agent (application client)	Sun Solaris	The scheduler is installed as part of the client installation
Tivoli Data Protection for Microsoft Exchange (application client) Tivoli Data Protection for SQL (application client) Tivoli Data Protection for Domino (application client) Tivoli Data Protection for Oracle (application client)	Windows	The scheduler is installed and configured separately
Tivoli Data Protection for Lotus Notes™ (application client)	Windows	Uses the Lotus Notes scheduler

Commands for Scheduling Client Operations

This section summarizes example commands that can be used for the scheduling tasks that are discussed in this chapter. See *Administrator's Reference* for server command details.

Define a schedule for a client:

```
define schedule engpoldom daily_backup starttime=21:00
duration=2 durunits=hours
```

Associate a client with a schedule:

```
define association engpoldom weekly_backup engnode
```

On the client workstation, start the scheduler:
On most clients:

```
> dsmc schedule
```

On Windows NT and Windows 2000 clients:

```
> net start "TSM Scheduler"
```

Check the *Backup-Archive Installation and User's Guide* for details about automatically starting the scheduler and running the scheduler in the background.

Display schedule information:

```
query schedule engpoldom
```

Check to see if the schedule ran successfully:

```
query event engpoldom weekly_backup begindate=-7 begintime=now
 enddate=today endtime=now
```

Chapter 15. Managing Schedules for Client Nodes

This chapter contains information about managing and coordinating Tivoli Storage Manager schedules for registered client nodes. For a description of what Tivoli Storage Manager views as client nodes, see “Chapter 10. Adding Client Nodes” on page 227. For information about the Tivoli Storage Manager scheduler and creating schedules, see “Chapter 14. Scheduling Operations for Client Nodes” on page 325.

Administrators can perform the following tasks:

Tasks:
“Managing Tivoli Storage Manager Schedules” on page 333
“Managing Node Associations with Schedules” on page 335
“Managing Event Records” on page 336
“Managing the Throughput of Scheduled Operations” on page 338
“Specifying One-Time Actions for Client Nodes” on page 344

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator’s Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Managing Tivoli Storage Manager Schedules

You can perform the following activities to manage schedules.

Task	**Required Privilege Class**
Verify that the schedule ran	Any administrator
Add, copy, modify, or delete client schedules in any policy domain	System or unrestricted policy
Add, copy, modify, or delete client schedules for specific policy domains	System, unrestricted policy, or restricted policy for those domains
Display information about scheduled operations	Any administrator

Verifying that the Schedule Ran

You will want to ensure that all defined nodes completed their scheduled operations. You can check whether the schedules ran successfully by using the QUERY EVENT command. For information, see “Displaying Information about Scheduled Events” on page 336.

You can also check the log file described in “Checking the Schedule Log” on page 334.

© Copyright IBM Corp. 1993, 2002

Checking the Schedule Log

The Tivoli Storage Manager client stores detailed information about each scheduled event in a file. This file contains information such as the statistics about the backed-up objects, the name of the server to which the objects are backed up, and the time and date of the next scheduled operation.

The default name for this file is *dsmsched.log*. The file is located in the directory where the Tivoli Storage Manager backup-archive client is installed. You can override this file name and location by specifying the SCHEDLOGNAME option in the client options file. See the client user's guide for more information.

Adding New Schedules

You can add new Tivoli Storage Manager schedules by using the DEFINE SCHEDULE command. After you add a new schedule, associate the node with the schedule. For more information, see "Defining Client Schedules" on page 326.

Copying Existing Schedules

You can create new schedules by copying existing schedules to the same policy domain or a different policy domain. The schedule description and all schedule parameter values are copied to the new schedule. You can then modify the new schedule to meet site-specific requirements.

Client node associations are not copied to the new schedule. You must associate client nodes with the new schedule before it can be used. The associations for the old schedule are not changed. For information, see "Associating Client Nodes with Schedules" on page 327.

To copy the WINTER schedule from policy domain DOMAIN1 to DOMAIN2 and name the new schedule WINTERCOPY, enter:

```
copy schedule domain1 winter domain2 wintercopy
```

Modifying Schedules

You can modify existing schedules by using the UPDATE SCHEDULE command.For example, to modify the ENGWEEKLY client schedule in the ENGPOLDOM policy domain, enter:

```
update schedule engpoldom engweekly period=5 perunits=days
```

The ENGWEEKLY schedule is updated so that the incremental backup period is now every five days.

Deleting Schedules

When you delete a schedule, Tivoli Storage Manager deletes all client node associations for that schedule. See "Associating Client Nodes with Schedules" on page 327 for more information.

To delete the schedule WINTER in the ENGPOLDOM policy domain, enter:

```
delete schedule engpoldom winter
```

Rather than delete a schedule, you may want to remove all nodes from the schedule and save the schedule for future use. For information, see "Removing Nodes from Schedules" on page 336.

Displaying Information about Schedules

When you request information about schedules, the server displays the following information:

- Schedule name
- Policy domain name
- Type of operation to be performed
- Start date and time for the initial startup window
- Duration of the startup window
- Time period between startup windows
- Day of the week on which scheduled operations can begin

The following output shows an example of a report that is displayed after you enter:

```
query schedule engpoldom
```

```
Domain       * Schedule Name    Action Start Date/Time      Duration Period Day
------------ - ---------------- ------ ------------------- -------- ------ ---
ENGPOLDOM      MONTHLY_BACKUP   Inc Bk 07/21/1998 12:45:14      2 H    2 Mo Sat
ENGPOLDOM      WEEKLY_BACKUP    Inc Bk 07/21/1998 12:46:21      4 H    1 W  Sat
```

Managing Node Associations with Schedules

You can add and delete node associations from schedules. Nodes can be associated with more than one schedule.

You can perform the following activities to manage associations of client nodes with schedules.

Task	**Required Privilege Class**
Add new nodes to existing schedules	System or restricted policy over the domain to which the node is assigned
Move nodes to existing schedules	System or restricted policy over the domain to which the node is assigned
Delete nodes associated with a schedule	System or restricted policy over the domain to which the node is assigned
Display nodes associated with a specific schedule	Any administrator

Adding New Nodes to Existing Schedules

You can add new nodes to existing schedules by associating the node with the schedule. To associate client nodes with a schedule, you can use the administrative Web interface or you can issue the DEFINE ASSOCIATION command from the command line interface. For information, see “Associating Client Nodes with Schedules” on page 327.

Moving Nodes from One Schedule to Another

You can move a node from one schedule to another schedule by:

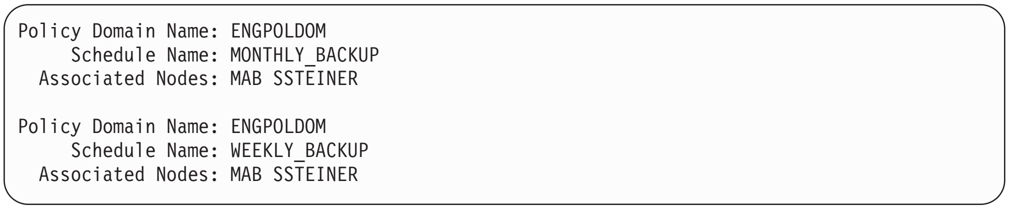
1. Associating the node to the new schedule. For information, see “Associating Client Nodes with Schedules” on page 327.
2. Deleting the association of that node from the original schedule. For information, see “Removing Nodes from Schedules” on page 336.

Displaying Nodes Associated with Schedules

You can display information about the nodes that are associated with a specific schedule. For example, you should query an association before deleting a client schedule.

Figure 49 shows the report that is displayed after you enter:

```
query association engpoldom
```



```
Policy Domain Name: ENGPOLDOM
     Schedule Name: MONTHLY_BACKUP
  Associated Nodes: MAB SSTEINER

Policy Domain Name: ENGPOLDOM
     Schedule Name: WEEKLY_BACKUP
  Associated Nodes: MAB SSTEINER
```

Figure 49. Query Association Output

Removing Nodes from Schedules

When you remove the association of a node to a client schedule, the client no longer runs operations specified by the schedule. However, the remaining client nodes still use the schedule.

To delete the association of the ENGNOD client with the ENGWEEKLY schedule, in the policy domain named ENGPOLDOM, enter:

```
delete association engpoldom engweekly engnod
```

Instead of deleting a schedule, you may want to delete all associations to it and save the schedule for possible reuse in the future.

Managing Event Records

Each scheduled client operation is called an *event*. All scheduled events, including their status, are tracked by the server. An *event record* is created in the server database whenever a scheduled event is completed or missed.

You can perform the following activities to manage event records:

Task	**Required Privilege Class**
Display information about scheduled events	Any administrator
Set the retention period for event records	System
Delete event records	System or unrestricted policy

Displaying Information about Scheduled Events

To help manage schedules for client operations, you can request information about scheduled and completed events by using the QUERY EVENT command.

- To get information about past and projected scheduled processes, use a simple query for events. If the time range you specify includes the future, the results show which events should occur in the future based on current schedules.

- To get information about scheduled processes that did not complete successfully, use the exceptions-only option with the query.

To minimize the processing time when querying events:

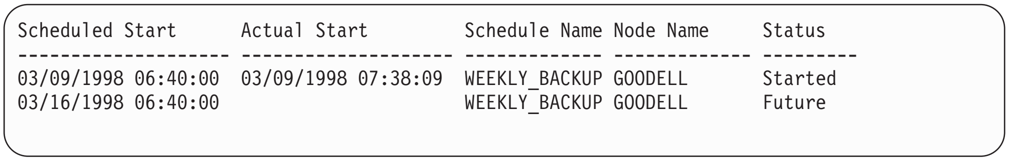
- Minimize the time range
- For client schedules, restrict the query to those policy domains, schedules, and client node names for which information is required

Displaying All Client Schedule Events

You can display information about all client events by issuing the QUERY EVENT command. The information includes events for both successful and failed schedules. If the administrator specifies a time range that includes the future, Tivoli Storage Manager displays future events with a status of *future*.

Figure 50 shows an example of a report for client node GOODELL that is displayed after you enter:

```
query event standard weekly_backup node=goodell enddate=today+7
```



```
Scheduled Start      Actual Start         Schedule Name Node Name     Status
-------------------- -------------------- ------------- ------------- ---------
03/09/1998 06:40:00  03/09/1998 07:38:09  WEEKLY_BACKUP GOODELL       Started
03/16/1998 06:40:00                       WEEKLY_BACKUP GOODELL       Future
```

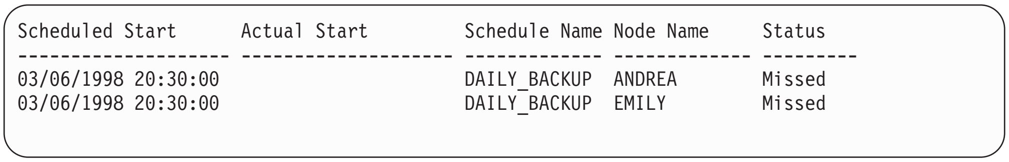
Figure 50. Events for a Node

Displaying Events that Ended Unsuccessfully

You can display information about scheduled events that ended unsuccessfully by using exception reporting. For example, you can issue the following command to find out which events were missed in the previous 24 hours, for the DAILY_BACKUP schedule in the STANDARD policy domain:

```
query event standard daily_backup begindate=-1 begintime=now
enddate=today endtime=now exceptionsonly=yes
```

Figure 51 shows an example of the results of this query. To find out why a schedule was missed or failed, you may need to check the schedule log on the client node itself. For example, a schedule can be missed because the scheduler was not started on the client node.



```
Scheduled Start      Actual Start         Schedule Name Node Name     Status
-------------------- -------------------- ------------- ------------- ---------
03/06/1998 20:30:00                       DAILY_BACKUP  ANDREA        Missed
03/06/1998 20:30:00                       DAILY_BACKUP  EMILY         Missed
```

Figure 51. Exception Report of Events

Displaying Past Events

If you query the server for events, the server may display past events even if the event records have been deleted. Such events are displayed with a status of *Uncertain*, indicating that complete information is not available because the event records have been deleted. To determine if event records have been deleted, check the message that is issued after the DELETE EVENT command is processed.

Managing Event Records in the Server Database

By default, the server retains event records for 10 days before automatically removing them from the database. The server automatically deletes event records from the database after the event retention period has passed and after the startup window for the event has elapsed.

You can specify how long event records stay in the database before the server automatically deletes them by using the SET EVENTRETENTION command. You can also manually delete event records from the database, if database space is required.

Setting the Event Retention Period

You can modify the retention period for event records in the database. To change the retention period to 15 days, enter:

```
set eventretention 15
```

Manually Deleting Event Records

You may want to manually delete event records to increase available database space. For example, to delete all event records written prior to 11:59 p.m. on June 30, 2000, enter:

```
delete event 06/30/2000 23:59
```

Managing the Throughput of Scheduled Operations

In the Tivoli Storage Manager environment where many nodes attempt to initiate scheduled operations simultaneously, you may have to manage scheduling throughput. You can choose a scheduling mode, and you can control how often client nodes contact the server to perform a scheduled operation.

Administrators can perform the following activities to manage the throughput of scheduled operations.

Task	Required Privilege Class
Modify the default scheduling mode	System
Modify the scheduling period for incremental backup operations	System
Balance the scheduled workload for the server	System
Set the frequency at which client nodes contact the server	System

Modifying the Default Scheduling Mode

Tivoli Storage Manager provides two scheduling modes: *client-polling* and *server-prompted*. The mode indicates how client nodes interact with the server for scheduling operations. With client-polling mode, client nodes poll the server for the next scheduled event. With server-prompted mode, the server contacts the nodes at the scheduled start time.

By default, the server permits both scheduling modes. The default (ANY) allows nodes to specify either scheduling mode in their client options files. You can modify this scheduling mode.

If you modify the default server setting to permit only one scheduling mode, *all* client nodes must specify the same scheduling mode in their client options file. Clients that do not have a matching scheduling mode will not process the scheduled operations. The default mode for client nodes is client-polling.

The scheduler must be started on the client node's machine before a schedule can run in either scheduling mode.

For more information about modes, see "Overview of Scheduling Modes".

Overview of Scheduling Modes

With client-polling mode, client nodes poll the server for the next scheduled event. With server-prompted mode, the server contacts the nodes at the scheduled start time. See Table 27 and Table 26.

Table 26. Client-Polling Mode

How the mode works	Advantages and disadvantages
1. A client node queries the server at prescribed time intervals to obtain a schedule. This interval is set with a client option, QUERYSCHEDPERIOD. For information about client options, refer to the appropriate *Backup-Archive Installation and User's Guide*. 2. At the scheduled start time, the client node performs the scheduled operation. 3. When the operation completes, the client sends the results to the server. 4. The client node queries the server for its next scheduled operation.	• Useful when a high percentage of clients start the scheduler manually on a daily basis, for example when their workstations are powered off nightly. • Supports *randomization*, which is the random distribution of scheduled start times. The administrator can control randomization. By randomizing the start times, Tivoli Storage Manager prevents all clients from attempting to start the schedule at the same time, which could overwhelm server resources. • Valid with all communication methods.

Table 27. Server-Prompted Mode

How the mode works	Advantages and disadvantages
1. The server contacts the client node when scheduled operations need to be performed and a server session is available. 2. When contacted, the client node queries the server for the operation, performs the operation, and sends the results to the server.	• Useful if you change the schedule start time frequently. The new start time is implemented without any action required from the client node. • Useful when a high percentage of clients are running the scheduler and are waiting for work. • Does not allow for randomization of scheduled start times. • Valid only with client nodes that use TCP/IP to communicate with the server.

Modifying the Scheduling Mode on the Server

If you modify the default so that the server permits only one scheduling mode for the server, all clients must specify the same scheduling mode in their client options file. Clients that do not have a matching scheduling mode do not process scheduled operations.

Client-Polling Scheduling Mode: To have clients poll the server for scheduled operations, enter:

```
set schedmodes polling
```

Ensure that client nodes specify the same mode in their client options files.

Server-Prompted Scheduling Mode: To have the server prompt clients for scheduled operations, enter:

```
set schedmodes prompted
```

Ensure that client nodes specify the same mode in their client options files.

Any Scheduling Mode: To return to the default scheduling mode so that the server supports both client-polling and server-prompted scheduling modes, enter:

```
set schedmodes any
```

Client nodes can then specify either polling or prompted mode.

Modifying the Default Scheduling Mode on Client Nodes

Users set the scheduling mode on client nodes. They specify either the client-polling or the server-prompted scheduling mode on the command line or in the client user options file. (On UNIX systems, root users set the scheduling mode in the client system options file.)

For more information, refer to the appropriate *Backup-Archive Installation and User's Guide*.

Specifying the Schedule Period for Incremental Backup Operations

When you define a backup copy group, you specify the copy frequency, which is the minimum interval between successive backups of a file. When you define a schedule, you specify the length of time between processing of the schedule. Consider how these interact to ensure that the clients get the backup coverage that you intend.

See "Defining and Updating a Backup Copy Group" on page 293.

Balancing the Scheduled Workload for the Server

You can control the server's workload and ensure that the server can perform all scheduled operations within the specified window. To enable the server to complete all schedules for clients, you may need to use trial and error to control the workload. To estimate how long client operations take, test schedules on several representative client nodes. Keep in mind, for example, that the first incremental backup for a client node takes longer than subsequent incremental backups.

You can balance the server's scheduled workload by:

- Adjusting the number of sessions that the server allocates to scheduled operations
- Randomizing scheduled start time for client operations (if clients use client-polling scheduling mode)
- Increasing the length of the startup window

Setting the Number of Sessions the Server Allocates to Scheduled Operations

The maximum number of concurrent client/server sessions is defined by the MAXSESSIONS server option. Of these sessions, you can set a maximum percentage to be available for processing scheduled operations. Limiting the number of sessions available for scheduled operations ensures that sessions are available when users initiate any unscheduled operations, such as restoring file or retrieving files.

If the number of sessions for scheduled operations is insufficient, you can increase either the total number of sessions or the maximum percentage of scheduled sessions. However, increasing the total number of sessions can adversely affect server performance. Increasing the maximum percentage of scheduled sessions can reduce the server availability to process unscheduled operations.

For example, assume that the maximum number of sessions between client nodes and the server is 80. If you want 25% of these sessions to be used by for scheduled operations, enter:

```
set maxschedsessions 25
```

The server then allows a maximum of 20 sessions to be used for scheduled operations.

The following table shows the tradeoffs of using either the SET MAXSCHEDSESSIONS command or the MAXSESSIONS server option.

An administrator can...	Using...	With the result
Increase the total number of sessions	MAXSESSIONS server option	May adversely affect the server’s performance
Increase the total number of sessions allocated to scheduled operations	SET MAXSCHEDSESSIONS command	May reduce the server’s ability to process unscheduled operations

For information about the MAXSESSIONS option and the SET MAXSCHEDSESSIONS command, refer to *Administrator’s Reference*.

Randomizing Schedule Start Times

To randomize start times for schedules means to scatter each schedule’s start time across its startup window. A startup window is defined by the start time and duration during which a schedule must be initiated. For example, if the start time is 1:00 a.m. and the duration is 4 hours, the startup window is 1:00 a.m. to 5:00 a.m.

For the client-polling scheduling mode, you can specify the percentage of the startup window that the server can use to randomize start times for different client nodes associated with a schedule.

If you set randomization to 0, no randomization occurs. This process can result in communication errors if many client nodes try to contact the server at the same instant.

The settings for randomization and the maximum percentage of scheduled sessions can affect whether schedules are successfully completed for client nodes. Users receive a message if all sessions are in use when they attempt to process a schedule. If this happens, you can increase randomization and the percentage of scheduled sessions allowed to make sure that the server can handle the workload. The maximum percentage of randomization allowed is 50%. This limit ensures that half of the startup window is available for retrying scheduled commands that have failed.

To set randomization to 50%, enter:

```
set randomize 50
```

It is possible, especially after a client node or the server has been restarted, that a client node may not poll the server until *after* the beginning of the startup window in which the next scheduled event is to start. In this case, the starting time is randomized over the specified percentage of the *remaining* duration of the startup window.

Consider the following situation:
- The schedule start time is 8:00 a.m. and its duration is 1 hour. Therefore the startup window for the event is from 8:00 to 9:00 a.m.
- Ten client nodes are associated with the schedule.
- Randomization is set to 50%.
- Nine client nodes poll the server before 8:00 a.m.
- One client node does not poll the server until 8:30 a.m.

The result is that the nine client nodes that polled the server *before* the beginning of the startup window are assigned randomly selected starting times between 8:00 and 8:30. The client node that polled at 8:30 receives a randomly selected starting time that is between 8:30 and 8:45.

Increasing the Length of the Schedule Startup Window

Increasing the size of the startup window (by increasing the schedule's duration) can also affect whether a schedule completes successfully. A larger startup window gives the client node more time to attempt initiation of a session with the server.

Controlling How Often Client Nodes Contact the Server

To control how often client nodes contact the server to perform a scheduled operation, an administrator can set:
- How often nodes query the server (see "Setting How Often Clients Query the Server" on page 343)
- The number of command retry attempts (see "Setting the Number of Command Retry Attempts" on page 343)
- The amount of time between retry attempts (see "Setting the Amount of Time between Retry Attempts" on page 343)

Users can also set these values in their client user options files. (Root users on UNIX systems set the values in client system options files.) However, user values are overridden by the values that the administrator specifies on the server.

The communication paths from client node to server can vary widely with regard to response time or the number of gateways. In such cases, you can choose *not* to set these values so that users can tailor them for their own needs.

Setting How Often Clients Query the Server

When scheduling client nodes with client-polling scheduling, you can specify how often the nodes query the server for a schedule. If nodes poll frequently for schedules, changes to scheduling information (through administrator commands) are propagated more quickly to the nodes. However, increased polling by client nodes also increases network traffic.

For the client-polling scheduling mode, you can specify the maximum number of hours that the scheduler on a client node waits between attempts to contact the server to obtain a schedule. You can set this period to correspond to the frequency with which the schedule changes are being made. If client nodes poll more frequently for schedules, changes to scheduling information (through administrator commands) are propagated more quickly to client nodes.

If you want to have all clients using polling mode contact the server every 24 hours, enter:

```
set queryschedperiod 24
```

This setting has no effect on clients that use the server-prompted scheduling mode.

The clients also have a QUERYSCHEDPERIOD option that can be set on each client. The server value overrides the client value once the client successfully contacts the server.

Setting the Number of Command Retry Attempts

You can specify the maximum number of times the scheduler on a client node can retry a scheduled command that fails.

The maximum number of command retry attempts does not limit the number of times that the client node can contact the server to obtain a schedule. The client node never gives up when trying to query the server for the next schedule.

Be sure not to specify so many retry attempts that the total retry time is longer than the average startup window.

If you want to have all client schedulers retry a failed attempt to process a scheduled command up to two times, enter:

```
set maxcmdretries 2
```

Maximum command retries can also be set on each client with a client option, MAXCMDRETRIES. The server value overrides the client value once the client successfully contacts the server.

Setting the Amount of Time between Retry Attempts

You can specify the length of time that the scheduler waits between command retry attempts. Command retry attempts occur when a client node is unsuccessful in establishing a session with the server or when a scheduled command fails to process. Typically, this setting is effective when set to half of the estimated time it takes to process an average schedule.

If you want to have the client scheduler retry every 15 minutes any failed attempts to either contact the server or process scheduled commands, enter:

```
set retryperiod 15
```

You can use this setting in conjunction with the SET MAXCMDRETRIES command (number of command retry attempts) to control when a client node contacts the server to process a failed command. See "Setting the Number of Command Retry Attempts" on page 343.

The retry period can also be set on each client with a client option, RETRYPERIOD. The server value overrides the client value once the client successfully contacts the server.

Specifying One-Time Actions for Client Nodes

You can use the DEFINE CLIENTACTION command to specify that one or more client nodes perform a one-time action if the client schedulers are active. If the scheduling mode is set to prompted, the client performs the action within 3 to 10 minutes. If the scheduling mode is set to polling, the client processes the command at its prescribed time interval. The time interval is set by the QUERYSCHEDPERIOD client option.

The DEFINE CLIENTACTION command causes Tivoli Storage Manager to automatically define a schedule and associate client nodes with that schedule. The schedule name and association information is returned to the server console or the administrative client with messages ANR2500I and ANR2510I. With the schedule name provided, you can later query or delete the schedule and associated nodes. The names of one-time client action schedules can be identified by a special character followed by numerals, for example @1.

For example, you can issue a DEFINE CLIENTACTION command that specifies an incremental backup command for client node HERMIONE in domain ENGPOLDOM:

```
define clientaction hermione domain=engpoldom action=incremental
```

Tivoli Storage Manager defines a schedule and associates client node HERMIONE with the schedule. The server assigns the schedule priority 1, sets the period units (PERUNITS) to ONETIME, and determines the number of days to keep the schedule active based on the value set with SET CLIENTACTDURATION command.

For a list of valid actions, see the DEFINE CLIENTACTION command in *Administrator's Reference*. You can optionally include the OPTIONS and OBJECTS parameters.

Determining How Long the One-Time Schedule Remains Active

You can determine how long schedules that were defined via DEFINE CLIENTACTION commands remain active by using the SET CLIENTACTDURATION command. This command allows you to specify the number of days that schedules that were created with the DEFINE CLIENTACTION command are active. These schedules are automatically removed from the database whether the associated nodes have processed the schedule or not, after the specified number of days. The following example specifies that schedules for client actions be active for 3 days:

```
set clientactduration 3
```

If the duration of client actions is set to zero, the server sets the DURUNITS parameter (duration units) as indefinite for schedules defined with DEFINE CLIENTACTION command. The indefinite setting for DURUNITS means that the schedules are not deleted from the database.

Part 4. Maintaining the Server

© Copyright IBM Corp. 1993, 2002

Chapter 16. Managing Server Operations

Administrators can perform such server operations as licensing purchased features, starting and halting the server, and monitoring server information. See the following sections:

Tasks:
"Licensing Tivoli Storage Manager"
"Starting and Halting the Server" on page 353
"Moving the Tivoli Storage Manager Server" on page 359
"Changing the Date and Time on the Server" on page 360
"Managing Server Processes" on page 360
"Preemption of Client or Server Operations" on page 361
"Setting the Server Name" on page 362
"Adding or Updating Server Options" on page 363
"Using Server Performance Options" on page 363
"Automatic Tuning of Server Options" on page 364
"Getting Help on Commands and Error Messages" on page 364

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Licensing Tivoli Storage Manager

This section describes the tasks involved when licensing a Tivoli Storage Manager system, including registering, saving and auditing.

Task	**Required Privilege Class**
Register licenses Audit licenses	System
Display license information	Any administrator

For current information about supported clients and devices, visit the Tivoli Storage Manager page on the World Wide Web at http://www.tivoli.com/support/storage_mgr/tivolimain.html.

The base Tivoli Storage Manager feature includes the following support:

- An unlimited number of administrative clients.
- Enterprise Administration, which includes: command routing, enterprise configuration, and enterprise logging (server-to-server).

© Copyright IBM Corp. 1993, 2002

- Server-to-server Virtual Volume capabilities (does not include database and storage pool backup).
- Network Enabler (network connections for clients).
- AFS/DFS Support, (the S/390® platform includes the S/390 UNIX client as part of Managed System for SAN).

Registering Licensed Features

You must register a new license if you want to add support for any of the following features that are not already in your existing license agreement. A license file and the REGISTER LICENSE command are used to complete this task. Licenses are stored in enrollment certificate files, which contain licensing information for the server product. The enrollment certificate files are on the installation CD-ROM. When registered, the licenses are stored in a NODELOCK file within the current directory.

Table 28. Licensed Features

License File Name	Description
mgsyslan.lic	Managed System for LAN
mgsyssan.lic	Managed System for SAN Required for each storage agent. The Tape Library Sharing feature is required on the Tivoli Storage Manager server. Tivoli Storage Manager requires this license for each client using server-free data movement.
spacemgr.lic	Each managed system that uses Tivoli Space Manager Also required: Managed System for LAN license if you are using a communication protocol other than shared memory. Only one Managed System for LAN license is required if an HSM client and backup-archive client are on the same system with the same node ID.
domino.lic	Each managed system that uses Tivoli Data Protection for Lotus Domino Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
emcsymm.lic	Each managed system that uses Tivoli Data Protection for EMC Symmetrix Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
emcsymr3.lic	Each managed system that uses Tivoli Data Protection for EMC Symmetrix R/3 Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
ess.lic	Each managed system that uses Tivoli Data Protection for ESS Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
essr3.lic	Each managed system that uses Tivoli Data Protection for ESS R/3 Also required: Managed System for LAN license if you use a communication protocol other than shared memory.

Table 28. Licensed Features (continued)

License File Name	Description
informix.lic	Each managed system that uses Tivoli Data Protection for Informix Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
lnotes.lic	Each managed system that uses Tivoli Data Protection for Lotus Notes Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
msexch.lic	Each managed system that uses Tivoli Data Protection for MS Exchange Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
mssql.lic	Each managed system that uses Tivoli Data Protection for MS SQL Server Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
oracle.lic	Each managed system that uses Tivoli Data Protection for Oracle Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
r3.lic	Each managed system that uses Tivoli Data Protection for R/3 Also required: Managed System for LAN license if you use a communication protocol other than shared memory.
library.lic	Managed Library Required for each library in the Extended Device Category that is managed by a Tivoli Storage Manager server. For current information on supported devices, visit the TSM page on the World Wide Web at http://www.tivoli.com/support/storage_mgr/tivolimain.html.
drm.lic	Tivoli Disaster Recovery Manager (includes server-to-server virtual volumes for database and storage pool backup) Required on a source server but not on a target server.
ndmp.lic	Tivoli Data Protection for NDMP Required on a Tivoli Storage Manager server that performs backup and restore operations of a NAS file server, using Network Data Management Protocol.
libshare.lic	Tape Library Sharing Required on a Tivoli Storage Manager server that can access a shared library, including the library manager. The Managed Library license is required only on the library manager.

To register a license, you must issue the REGISTER LICENSE command as well as the license file associated with the license. For example, to use the Tivoli Disaster Recovery Manager and two Tivoli Space Manager (HSM clients), issue the following commands:

```
register license file=drm.lic
register license file=spacemgr.lic number=2
register license file=mgsyslan.lic number=2
```

To register 20 managed systems that move data over a local area network, issue the following command:

```
register license file=mgsyslan.lic  number=20
```

To register 10 Tivoli Data Protection for Lotus Notes clients that move data over a LAN, using the TCP/IP communication protocol, issue the following commands:

```
register license file=lnotes.lic number=10
register license file=mgsyslan.lic number=10
```

With the exception of the ndmp.lic, drm.lic and libshare.lic, you can specify any number of license files to register. Always specify the total number of licenses you want registered. The REGISTER LICENSE command updates the nodelock file based on the total number of licenses you want registered. If you enter number=0 for a particular license, the license is unregistered. If you have twenty licenses and require ten additional licenses, you must register thirty.

For example, to register Tivoli Storage Manager for 30 managed systems that move data over a local area network, issue the following commands:

```
register license file=mgsyslan.lic number=30
```

You can also register a license by specifying the product password that is included in the license certificate file. For example:

```
register license 5s3qydpnwx7njdxnafksqas4
```

Attention: Tivoli Storage Manager licenses are associated with the CPU chip of the machine on which Tivoli Storage Manager is installed. If you change that CPU chip, you must first erase the existing nodelock files and then reregister all your licenses.

Saving Your Licenses

Save the CD-ROM containing your enrollment certificate files. You may you need to register your licenses again for any of the following reasons:

- The server is corrupted.
- The server is moved to a different machine.
- The *Nodelock* file is destroyed or corrupted. Tivoli Storage Manager stores license information in the *Nodelock* file, which is located in the directory from which the server is started.

Monitoring Licenses

When license terms change (for example, a new license is specified for the server), the server conducts an audit to determine if the current server configuration conforms to the license terms. The server also periodically audits compliance with license terms. The results of an audit are used to check and enforce license terms. If 30 days have elapsed since the previous license audit, the administrator cannot cancel the audit.

If a Tivoli Storage Manager system exceeds the terms of its license agreement, one of the following occurs:

- The server issues a warning message indicating that it is not in compliance with the licensing terms.

- If you are running in Try Buy mode, operations fail because the server is not licensed for specific features.

You must contact your Tivoli Storage Manager account representative or authorized reseller to modify your agreement.

An administrator can monitor license compliance by:

Auditing licenses
Use the AUDIT LICENSES command to compare the current configuration with the current licenses.

Note: During a license audit, the server calculates, by node, the amount of backup, archive, and space management storage in use. This calculation can take a great deal of CPU time and can stall other server activity. Use the AUDITSTORAGE server option to specify that storage is not to be calculated as part of a license audit.

Displaying license information
Use the QUERY LICENSE command to display details of your current licenses and determine licensing compliance.

Scheduling automatic license audits
Use the SET LICENSEAUDITPERIOD command to specify the number of days between automatic audits.

Starting and Halting the Server

Task	Required Privilege Class
Start, halt, and restart the server	System or operator

Starting the Server

The following events occur when you start or restart the Tivoli Storage Manager server:

- The server invokes the communication methods specified in the server options file.
- The server uses the volumes specified in the dsmserv.dsk file for the database and recovery log to record activity. It also identifies storage pool volumes to be used.
- The server starts a Tivoli Storage Manager server console session that is used to operate and administer the server until administrative clients are registered to the server.

To start the server, complete the following steps:

1. Change to the /usr/tivoli/tsm/server/bin directory from an AIX session. Enter:
   ```
   cd /usr/tivoli/tsm/server/bin
   ```
2. Start the server by entering:
   ```
   dsmserv
   ```

   Note: If the server does not start, set the ulimit parameter to unlimited. For example,
   ```
   ulimit -d unlimited
   ```

When the server is started, Tivoli Storage Manager displays the following information:

- Product licensing and copyright information
- Processing information about the server options file
- Communication protocol information
- Database and recovery log information
- Storage pool volume information
- Server generation date
- Progress messages and any errors encountered during server initialization

If Tivoli Storage Manager detects an invalid system date and time, the server is disabled, and expiration, migration, reclamation, and volume history deletion operations are not allowed. An error message (ANR0110E) is displayed. You may either change the system date if it is in error, or issue the ACCEPT DATE command to force the server to accept the current system date as valid. After the system date is resolved, you must issue the ENABLE SESSIONS command to re-enable the server for client sessions.

The date and time check occur when the server is started and once each hour thereafter. An invalid date is one that is:

- Earlier than the server installation date and time
- More than one hour earlier than the last time the date was checked
- More than 30 days later than the last time the date was checked

Running the Server in Background Mode

You may choose to run the server in the background. When the server runs in the background, you control the server through your administrative client.

Attention: Before running the server in the background, ensure the following conditions exist:

1. An administrative node has been registered and granted system authority. See "Registering Administrators" on page 261.
2. The administrative client options file has been updated with the correct SERVERNAME and TCPPORT options.
3. The administrative client can access the Tivoli Storage Manager server.

If you do not follow these steps, you cannot control the server. When this occurs, you can only stop the server by canceling the process, using the process number displayed at startup. You may not be able to take down the server cleanly without this process number.

To start the server running in the background, enter the following:

```
nohup dsmserv quiet &
```

You can check your directory for the output created in the nohup.out file to determine if the server has started. This file can grow considerably over time.

Capturing Server Console Messages to a User Log File

You can capture Tivoli Storage Manager server console messages to a user log file with the Tivoli Storage Manager dsmulog utility. You can invoke the utility with the ADSMSTART shell script which is provided as part of the Tivoli Storage Manager AIX server package. You can have the server messages written to one or more user log files. When the dsmulog utility detects that the server it is capturing messages from is stopped or halted, it closes the current log file and ends its processing.

When you specify more than one file, Tivoli Storage Manager manages the user logs as a circular list of files based on size or change of day. You can manage the amount of space the logs used in the file system by specifying a size parameter (in kilobytes) in the ADSMSTART shell script for the dsmulog utility. When the specified limit is reached, Tivoli Storage Manager closes the current user log and opens the next user log. When the specified limit is reached on the next user log, Tivoli Storage Manager writes to the next user log and can overwrite the previous contents of the file. If a size parameter is not specified, the utility writes to the next user log file when it detects a change of day.

If the user log file names are not fully qualified in the ADSMSTART shell script, the user logs are created in the directory where ADSMSTART is invoked. The user logs should not be placed in the /usr/lpp file system because space constraints in the file system can prevent the Tivoli Storage Manager server from starting.

The following is an example of how to set up and invoke the dsmulog utility to rotate through the user logs on a daily basis:

1. Change to the server bin directory:
   ```
   cd /usr/tivoli/tsm/server/bin
   ```
2. Copy ADSMSTART.SMP to ADSMSTART:
   ```
   cp adsmstart.smp ./adsmstart
   ```
3. Edit ADSMSTART. Do NOT change the first line in the file. Specify the user log files to capture messages on a daily basis. For example:
   ```
   dsmulog /u/admin/log1 /u/admin/log2 /u/admin/log3
   ```

   The following steps automatically start the server with console logging when the system is rebooted:
4. If the server is running, halt the server.
5. Run the *dsm_rmv_itab autostart* script.
6. Run the *dsm_update_itab autotrace* script.
7. Restart the server in one of the following ways:
   - If you restart the server by running the ADSMSTART script, the server runs in the foreground and all console output is sent to the specified user logs.
   - If you restart the server by issuing *nohup adsmstart &*, the server runs in the background and all console output is sent to the specified user logs. You must then use an administrative client session to halt the server.

In the above example, if you invoke the utility on Friday, on Friday the server messages are captured to log1, on Saturday the messages are captured to log2, and on Sunday the messages are captured to log3. On Monday the messages are captured to log1 and the previous Friday messages are overwritten.

The following example shows how to invoke the dsmulog utility to rotate through the user logs based on size limit:

```
dsmulog /u/admin/log1 /u/admin/log2 /u/admin/log3 size=500
```

When the server is started, the utility captures the server messages to log1 until it reaches a file size of 500 kilobytes and then changes to log2.

Tip: If the Tivoli Storage Manager server goes down unexpectedly, copy the current user logs to other file names before you restart the server. This will prevent the dsmulog utility from overwriting the current logs. You can then view the user logs to try and determine the cause of the unavailability of the server.

To log console messages during the current session, do the following:

1. If the server is running, halt the server.
2. Issue the dsmserv command as specified in the ADSMSTART shell script. For example:
   ```
   /usr/tivoli/tsm/server/bin/dsmserv 2>&1 | dsmulog /u/admin/log1 /u/admin/log2
   ```

To stop console logging and have the server automatically start after a system reboot, complete the following steps:

1. If the server is running, halt the server.
2. Change to the server bin directory:
   ```
   cd /usr/tivoli/tsm/server/bin
   ```
3. Run the *dsm_rmv_itab autotrace* script.
4. Run the *dsm_update_itab autostart* script.
5. Restart the server by running the *rc.adsmserv* script. This script starts the server in the quiet mode.

Starting the Server in Other Modes

The following Tivoli Storage Manager command options specify how you can start the server in other modes as part of the dsmserv command. For example:

```
dsmserv option
```

Where *option* can be any one of the following:

quiet Starts the server as a daemon program. The server runs as a background process, and does not read commands from the server console. Output messages are directed to the SERVER_CONSOLE.

> **Note:** Before issuing this command, you must have an administrative client registered and authorized with system authority. The administrative client must be started. Otherwise, the server will run in the quiet mode and you will not be able to access the server.

-o *filename*
Specifies an explicit options file name when running more than one server.

Defining Environment Variables

If you want to run the Tivoli Storage Manager server from a directory other than the default directory or to run multiple servers, you may have to define environment variables.

An *environment variable* describes the operating environment of a process, such as the home directory or the terminal in use. It provides the path that the server requires to find and create files.

For example, to define the DSMSERV_DIR environment variable to point to the usr/lpp/adsmserv/bin directory so that the server can find various files, such as dsmreg.lic or the message file (*dsmameng.txt*) enter:

```
export DSMSERV_DIR=/usr/tivoli/tsm/server/bin
```

You can also define an environment variable to point to the server options file. For example, to define the DSMSERV_CONFIG environment variable to point to the server options file, enter:

```
export DSMSERV_CONFIG=/usr/tivoli/tsm/server/bin/ filename.opt
```

where *filename* is the name you assigned your server options file (dsmserv.opt).

Notes:

1. The -o parameter of the DSMSERV command can also be used to specify an options file name.
2. The *set environment* command:

   ```
   setenv DSMSERV_DIR /usr/tivoli/tsm/server/bin
   ```

   is issued if your shell is in the "csh" family.
3. If you want to save this environment, save these entries in the *.kshrc* or the *.cshrc* file of your $HOME directory.
4. The dsmserv.dsk is always read from the directory in which the server is started.

Running Multiple Servers on a Single Machine

To have multiple servers running on a single machine, issue the DSMSERV FORMAT command from different directories to create multiple pairs of recovery log and database files. You do not have to install the server executable files in more than one directory.

However, if non-root users will be running servers, you must modify the access permission by adding read permission to the following files:

- dsmlicense
- dsmtli.drv

Use these commands as a root user to modify the permission for these files:

```
chmod 755 /usr/tivoli/tsm/server/bin/dsmlicense
chmod 755 /usr/tivoli/tsm/server/bin/dsmtli.drv
```

The following procedure shows how to set up an additional Tivoli Storage Manager server:

1. Determine the directory where you want the server files created, for example, /usr/tivoli/tsm/myserver, and make that directory:

   ```
   mkdir /usr/tivoli/tsm/myserver
   ```
2. Copy the dsmserv.opt file to your directory:

   ```
   cp /usr/tivoli/tsm/server/bin/dsmserv.opt
   dsmserv.opt /usr/tivoli/tsm/myserver/dsmserv.opt
   ```

   Note: Ensure that the communication parameters are unique among all other Tivoli Storage Manager servers. The communication protocols are:

   - TCPPORT for TCP/IP
   - HTTPPORT for HTTP Access in the Web Administrative Client Browser

   For example, if your first server is using the default TCPport of 1500, ensure that the new server is using a TCPport other than 1500 by providing a real value in the server options file.

3. Set your path on the server console or from an aixterm session. Define your environment variables, for example:

   To define the DSMSERV_DIR, enter:

   ```
   export DSMSERV_DIR=/usr/tivoli/tsm/server/bin
   ```

   Ensure that you are in the target directory before continuing.
4. Format the database and recovery log files, for example:

   ```
   /usr/tivoli/tsm/server/bin/dsmfmt -m -db dbvol2 5
   /usr/tivoli/tsm/server/bin/dsmfmt -m -log logvol2 9
   ```

   In this example, db indicates the database log, -m indicates megabytes and log indicates the recovery log. Refer to *Administrator's Reference* for more information on these commands.
5. Create the database and recovery log in the desired directory for the new server, for example:

   ```
   /usr/tivoli/tsm/server/bin/dsmserv format 1 logvol2 1 dbvol2
   ```

   Note: You need additional license authorizations to run additional servers. You can use the register license file command to register these licenses. See "Registering Licensed Features" on page 350 for more information.

Halting the Server

You can halt the server without warning if an unplanned operating system problem requires the server to be stopped.

When you halt the server, all processes are abruptly stopped and client sessions are canceled, even if they are not complete. Any in-progress transactions are rolled back when the server is restarted. Administrator activity is not possible.

If possible, halt the server only after current administrative and client node sessions have completed or canceled. To shut down the server without severely impacting administrative and client node activity with the server, you must:

1. Disable the server to prevent new client node sessions from starting by issuing the DISABLE SESSIONS command. This command does not cancel sessions currently in progress or system processes like migration and reclamation.
2. Notify any existing administrative and client node sessions that you plan to shut down the server. The server does not provide a network notification facility; you must use external means to notify users.
3. Cancel any existing administrative or client node sessions by issuing the CANCEL SESSION command and the associated session number. To obtain session numbers and determine if any sessions are running, use the QUERY SESSION command. If a session if running, a table will appear showing the session number on the far left side of the screen.
4. Find out if any other processes are running, such as server migration or inventory expiration, by using the QUERY PROCESS command. If a database backup process is running, allow it to complete before halting the server. If other types of processes are running, cancel them by using the CANCEL PROCESS command.

   Note: If the process you want to cancel is currently waiting for a tape volume to be mounted (for example, a process initiated by EXPORT, IMPORT, or MOVE DATA commands), the mount request is automatically cancelled.

If a volume associated with the process is currently being mounted by an *automated* library, the cancel may not take effect until the mount is complete.

5. Halt the server to shut down all server operations by using the HALT command.

Note: The QUIESCE option on the HALT command is recommended *only* if you plan to do a database dump by using the DSMSERV DUMPDB command immediately after halting. Because Tivoli Storage Manager supports online database backup (BACKUP DB command), the DSMSERV DUMPDB command should be rarely, if ever, needed.

Stopping the Server When Running as a Background Process

If you started the server as a background process and want to stop the server, connect to the server as an administrative client and issue the HALT command. If you cannot connect to the server with an administrative client and you want to stop the server, you must cancel the process by using the **kill** command with the process ID number (pid) that is displayed at initialization.

Note: Before you issue the **kill** command, ensure that you know the correct process ID for the Tivoli Storage Manager server.

Moving the Tivoli Storage Manager Server

There are two ways to move a Tivoli Storage Manager server to a new machine that is running the same operating system:

- Back up the database, and restore it on another machine on which Tivoli Storage Manager is installed. The backup and restore method is described in this section.
- Export the server to sequential media, and import the data to the new machine. See “Chapter 21. Exporting and Importing Data” on page 477 for details. The export and import method may take longer than the backup and restore method.

The prerequisites for the backup and restore method are:

- The same operating system must be running on both machines.
- The sequential storage pool must be accessible from both machines. You will have to move any libraries and devices from the original machine to the new machine or they must be accessible through a SAN.
- The restore operation must be done by a Tivoli Storage Manager server at a code level that is the same as or later than that on the machine that was backed up.
- Only manual and SCSI library types are supported for the restore operation.

In the following example, the Tivoli Storage Manager server on machine A is moved to machine B.

On machine A:

1. Migrate all disk storage pool data to sequential media. See “Migration for Disk Storage Pools” on page 176 for details.
2. Perform a full database backup to sequential media.

   ```
   backup db devclass=8mm type=full
   ```
3. Copy the volume history file and the device configuration file.

On machine B:

4. Install Tivoli Storage Manager.
5. Copy the volume history file and device configuration file to the new server.
6. Restore the database:

```
dsmserv restore db devclass=8mm volumenames=vol001,vol002,vol002
```

Changing the Date and Time on the Server

Every time the server is started and for each hour thereafter, a date and time check occurs. An invalid date can be one of the following:

- Earlier than the server installation date and time.
- More than one hour earlier than the last time the date was checked.
- More than 30 days later than the last time the date was checked.

If the server detects an invalid date or time, server sessions become disabled. An error message (ANR0110E) is displayed and expiration, migration, reclamation, and volume history deletion operations are not allowed. You may either change the system date if it is in error, or issue the ACCEPT DATE command to force the server to accept the current system date as valid. Use the ENABLE SESSIONS command after you issue the ACCEPT DATE command to re-enable the server for client node activity.

Managing Server Processes

Task	**Required Privilege Class**
Display information about a server background process	Any administrator
Cancel a server process	System

When a user or administrator issues a Tivoli Storage Manager command or uses a graphical user interface to perform an operation, the server starts a process. Some examples of an operation are registering a client node, deleting a management class, or canceling a client session.

Most processes occur quickly and are run in the foreground, but others that take longer to complete run as background processes.

The server runs the following operations as background processes:

- Auditing an automated library
- Auditing licenses
- Auditing a volume
- Backing up the database
- Backing up a storage pool
- Checking volumes in and out of an automated library
- Defining a database volume copy
- Defining a recovery log volume copy
- Deleting a database volume
- Deleting a file space
- Deleting a recovery log volume
- Deleting a storage volume
- Expiring the inventory
- Exporting or importing data
- Extending the database or recovery log
- Generating a backup set
- Migrating files from one storage pool to the next storage pool
- Moving data from a storage volume
- Reclaiming space from tape storage volumes

- Reducing the database or recovery log
- Restoring a storage pool
- Restoring a volume
- Varying a database or recovery log volume online

The server assigns each background process an ID number and displays the process ID when the operation starts. This process ID number is used for tracking purposes. For example, if you issue an EXPORT NODE command, the server displays a message similar to the following:

```
EXPORT NODE started as Process 10
```

Some of these processes can also be run in the foreground by using the WAIT=YES parameter when you issue the command from an administrative client. See *Administrator's Reference* for details.

Requesting Information about Server Processes

You can request information about server background processes. If you know the process ID number, you can use the number to limit the search. However, if you do not know the process ID, you can display information about all background processes by entering:

```
query process
```

The following figure shows a server background process report after a DELETE FILESPACE command was issued. The report displays a process ID number, a description and a completion status for each background process.

```
 Process Process Description      Status
  Number
-------- ------------------------ ---------------------------------------------
       2 DELETE FILESPACE         Deleting filespace DRIVE_D for node CLIENT1:
                                   172 files deleted.
```

Canceling Server Processes

You can cancel a server background process by specifying its ID number in the following command:

```
cancel process 2
```

You can issue the QUERY PROCESS command to find the process number. See "Requesting Information about Server Processes" for details.

If the process you want to cancel is currently waiting for a tape volume to be mounted (for example, a process initiated by EXPORT, IMPORT, or MOVE DATA commands), the mount request is automatically canceled. If a volume associated with the process is currently being mounted by an *automated* library, the cancel may not take effect until the mount is complete.

Preemption of Client or Server Operations

The server can preempt server or client operations for a higher priority operation when a mount point is in use and no others are available, or access to a specific volume is required.

Mount Point Preemption

The following are high priority operations that can preempt operations for a mount point:

- Backup database
- Restore
- Retrieve
- HSM recall
- Export
- Import

The following lists operations that can be preempted and are listed in order of priority. The server selects the lowest priority operation to preempt, for example reclamation.

1. Move data
2. Migration from disk to sequential media
3. Backup, archive, or HSM migration
4. Migration from sequential media to sequential media
5. Reclamation

You can disable preemption by specifying NOPREEMPT in the server options file. When this option is specified, the BACKUP DB command is the only operation that can preempt other operations.

Volume Access Preemption

A high priority operation that requires access to a specific volume currently in use by a low priority operation can automatically preempt the operation. For example, if a restore request requires access to a volume in use by a reclamation process and a drive is available, the reclamation process is canceled and message ANR0494I or ANR1441I is issued.

The following are high priority operations that can preempt operations for access to a specific volume:

- Restore
- Retrieve
- HSM recall

The following lists operations that can be preempted, and are listed in order of priority. The server preempts the lowest priority operation, for example reclamation.

1. Move data
2. Migration from disk to sequential media
3. Backup, archive, or HSM migration
4. Migration from sequential media
5. Reclamation

You can disable preemption by specifying NOPREEMPT in the server options file. When this option is specified, no operation can preempt another operation for access to a volume.

Setting the Server Name

Task	Required Privilege Class
Specify the server name	System

At installation, the server name is set to SERVER1. After installation, you can use the SET SERVERNAME command to change the server name. You can use the QUERY STATUS command to see the name of the server.

To specify the server name as WELLS_DESIGN_DEPT., for example, enter the following:

```
set servername wells_design_dept.
```

You must set unique names on servers that communicate with each other. See "Setting Up Communications Among Servers" on page 436 for details.

Adding or Updating Server Options

Task	Required Privilege Class
Add or update a server option	System

You can add or update server options by editing the dsmserv.opt file, using the SETOPT command. For information about editing the server options file, refer to *Administrator's Reference*.

Adding or Updating a Server Option without Restarting the Server

A system administrator can add or update a limited number of server options without stopping and restarting the server. The added or updated server option is appended to the end of the server options file.

The following example shows how to use the SETOPT command to update the existing server option for MAXSESSIONS:

```
setopt maxsessions 20
```

The following lists server options that can be added or updated:

- BUFPOOLSIZE
- COMMTIMEOUT
- EXPINTERVAL
- EXPQUIET
- IDLETIMEOUT
- MAXSESSIONS
- THROUGHPUTDATATHRESHOLD
- THROUGHPUTTIMETHRESHOLD

Note: SETOPT commands in a macro cannot be rolled back.

Using Server Performance Options

Tivoli Storage Manager servers on AIX workstations, with an operating system level of 4.3.3 or greater and the asynchronous input/output (AIO) subsystem enabled, can take advantage of server performance options. These options enhance the data transfer capabilities to and from eligible Tivoli Storage Manager database volumes and log volumes in a disk storage pool. We recommend that you refer to the AIX documentation for additional information regarding the AIO subsystem and requirements. The performance options are:

- AIXASYNCIO

  To enable the server to use the AIX Asynchronous I/O support, set the AIXASYNCIO option to Yes. Asynchronous I/O support enhances system performance by allowing individual I/O requests to be gathered into one request and written to disk in parallel. To disable AIXASYNCIO support, set the option to NO. The default is for AIXASYNCIO not to be enabled.
- AIXDIRECTIO

  The AIXDIRECTIO option is enabled by default or by specifying YES. To disable the direct I/O option, set the option to NO. Direct I/O enhances system performance by allowing write activities to skip caching and go directly to disk.

Automatic Tuning of Server Options

For optimal performance, the server can tune the following server options automatically:

- MOVEBATCHSIZE and MOVESIZETHRESH

  To have the server automatically tune the MOVEBATCHSIZE and MOVESIZETHRESH options, set the SELFTUNETXNSIZE option to Yes. When the server performs an internal data movement operation, such as migration, reclamation, move data, storage pool backup or restore, it will adjust these values to achieve optimal performance. To prevent running out of log space during these operations, use the DEFINE SPACETRIGGER command to allow for expansion of the recovery log.
- BUFPOOLSIZE

  To have the server automatically tune the BUFPOOLSIZE option, set the SELFTUNEBUFPOOLSIZE option to Yes. Before expiration processing, the server resets the database buffer pool and examines the database buffer pool cache hit ratio. The server accounts for the amount of available storage and adjusts the buffer pool size as needed.

Note: Although the values of TXNGROUPMAX, MOVEBATCHSIZE, and MOVESIZETHRESH may be changed, the settings in the server options file are not changed. Issuing a QUERY OPTION command displays only what is set in the server options file.

For information about the SELFTUNEBUFPOOLSIZE and SELFTUNETXNSIZE server options, refer to *Administrator's Reference*.

Getting Help on Commands and Error Messages

Any administrator can issue the HELP command to display information about administrative commands and messages from the server and the administrative command-line client. You can issue the HELP command with no operands to display a menu of help selections. You also can issue the HELP command with operands that specify help menu numbers, commands, or message numbers.

To display the help menu, enter:

```
help
```

To display help information on the REMOVE commands, enter:

```
help remove
```

To display help information on a specific message, such as ANR0992I for example, enter:

```
help 0992
```

Additional information is also available in the online documentation.

Chapter 17. Automating Server Operations

You can schedule administrative commands to tune server operations and to start functions that require significant server or system resources during times of low usage. Automating these operations allows the administrator to ensure that server resources are available when needed by clients.

An administrator can automate the process of issuing a sequence of commands by storing the commands in a server script. From the command line, the administrator can immediately process the script or schedule the script for processing.

Tivoli Storage Manager includes a central scheduling component that allows the automatic processing of administrative commands during a specific time period when the schedule is activated. Schedules that are started by the scheduler can run in parallel. You can process scheduled commands sequentially by using Tivoli Storage Manager scripts that contain a sequence of commands with WAIT=YES. You can also use a scheduler external to Tivoli Storage Manager to invoke the administrative client to start one or more administrative commands.

Each scheduled administrative command is called an *event*. The server tracks and records each scheduled event in the database. You can delete event records as needed to recover database space.

See the following sections:

Tasks:
"Automating a Basic Administrative Command Schedule"
"Tailoring Schedules" on page 369
"Copying Schedules" on page 371
"Deleting Schedules" on page 371
"Managing Scheduled Event Records" on page 371
"Tivoli Storage Manager Server Scripts" on page 372
"Using Macros" on page 379

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Automating a Basic Administrative Command Schedule

This section describes how to set up a basic administrative command schedule using Tivoli Storage Manager defaults. To later update or tailor your schedules, see "Tailoring Schedules" on page 369.

© Copyright IBM Corp. 1993, 2002

Notes:

1. Scheduled administrative command output is directed to the activity log. This output cannot be redirected. For information about the length of time activity log information is retained in the database, see “Using the Tivoli Storage Manager Activity Log” on page 411.
2. You cannot schedule MACRO or QUERY ACTLOG commands.

Task	Required Privilege Class
Define, update, copy, or delete administrative schedules	System
Display information about scheduled operations	Any administrator

Defining the Schedule

Use the DEFINE SCHEDULE command to create a new schedule for processing an administrative command. Include the following parameters:

- Specify the administrative command to be issued (CMD=).
- Specify whether the schedule is activated (ACTIVE=).

For example:

```
define schedule backup_archivepool type=administrative
cmd='backup stgpool archivepool recoverypool' active=yes
```

This command results in the following:

- The schedule created is *BACKUP_ARCHIVEPOOL*.
- The schedule is to process the administrative command:

  ```
  backup stgpool archivepool recoverypool
  ```

  This command specifies that primary storage pool ARCHIVEPOOL is backed up to the copy storage pool RECOVERYPOOL.
- The schedule is currently active.
- Administrative command output is redirected to the activity log.
- The following defaults are in effect:
  - The start date and time defaults to the current date and time.
  - The length of the startup window is 1 hour.
  - The priority for the schedule is 5. If schedules conflict, the schedule with the highest priority (lowest number) is run first.
  - The schedule never expires.

To change the defaults, see “Tailoring Schedules” on page 369.

Verifying the Schedule

You can verify the details of what you have scheduled by using the QUERY SCHEDULE command. When you use the QUERY SCHEDULE command, you must specify the TYPE=ADMINISTRATIVE parameter to view an administrative command schedule. The following figure shows an example of a report that is displayed after you enter:

```
query schedule backup_archivepool type=administrative
```

```
*   Schedule Name       Start Date/Time       Duration   Period   Day
-   ----------------    --------------------  --------   ------   ---
     BACKUP_ARCHIVE-    03/15/1998 14:08:11        1 H      1 D   Any
     POOL
```

Note: The asterisk (*) in the first column specifies whether the corresponding schedule has expired. If there is an asterisk in this column, the schedule has expired.

You can check when the schedule is projected to run and whether it ran successfully by using the QUERY EVENT command. For information about querying events, see “Querying Events” on page 371.

Tailoring Schedules

To control more precisely when and how your schedules run, specify values for schedule parameters instead of accepting the defaults when you define or update schedules.

Schedule name
All schedules must have a unique name, which can be up to 30 characters.

Initial start date, time, and day
You can specify a past date, the current date, or a future date for the initial start date for a schedule with the STARTDATE parameter.

You can specify a start time, such as 6 p.m. with the STARTTIME parameter.

You can also specify the day of the week on which the startup window begins with the DAYOFWEEK parameter. If the start date and start time fall on a day that does not correspond to your value for the day of the week, the start date and time are shifted forward in 24-hour increments until the day of the week is satisfied.

If you select a value for the day of the week other than ANY, schedules may not process when you expect. This depends on the values for PERIOD and PERUNITS. Use the QUERY EVENT command to project when schedules will process to ensure that you achieve the desired result.

Duration of a startup window
You can specify the duration of a startup window, such as 12 hours, with the DURATION and DURUNITS parameters. The server must start the scheduled service within the specified duration, but does not necessarily complete it within that period of time. If the schedule needs to be retried for any reason, the retry attempt must begin before the startup window elapses or the operation does not restart.

If the schedule does not start during the startup window, the server records this as a *missed event* in the database. You can get an exception report from the server to identify schedules that did not run. For more information, see “Querying Events” on page 371.

How often to run the scheduled service
You can set the schedule frequency based on a period of hours, days, weeks, months, or years with the PERIOD and PERUNITS parameters. To have weekly backups, for example, set the period to one week with PERIOD=1 and PERUNITS=WEEKS.

Expiration date

You can specify an expiration date for a schedule with the EXPIRATION parameter if the services it initiates are required for only a specific period of time. If you set an expiration date, the schedule is not used after that date, but it still exists. You must delete the schedule to remove it from the database.

Priority

You can assign a priority to schedules with the PRIORITY parameter. For example, if you define two schedules and they have the same startup window or windows overlap, the server runs the schedule with the highest priority first. A schedule with a priority of 1 is started before a schedule with a priority of 3.

If two schedules try to use the same resources, the schedule that first initiated the process will be the one to continue processing. The second schedule will start but will not successfully complete. Be sure to check the activity log for details.

Administrative schedule name

If you are defining or updating an administrative command schedule, you **must** specify the schedule name.

Type of schedule

If you are updating an administrative command schedule, you **must** specify TYPE=ADMINISTRATIVE on the UPDATE command. If you are defining a new administrative command schedule, this parameter is assumed if the CMD parameter is specified.

Command

When you define an administrative command schedule, you **must** specify the complete command that is processed with the schedule with the CMD parameter. These commands are used to tune server operations or to start functions that require significant server or system resources. The functions include:

- Migration
- Reclamation
- Export and import
- Database backup

Whether or not the schedule is active

Administrative command schedules can be active or inactive when they are defined or updated. Active schedules are processed when the specified command window occurs. Inactive schedules are not processed until they are made active by an UPDATE SCHEDULE command with the ACTIVE parameter set to YES.

Example: Defining and Updating an Administrative Command Schedule

To schedule the backup of the ARCHIVEPOOL primary storage pool, enter:

```
define schedule backup_archivepool type=administrative
cmd='backup stgpool archivepool recoverypool'
active=yes startime=20:00 period=2
```

This command specifies that, starting today, the ARCHIVEPOOL primary storage pool is to be backed up to the RECOVERYPOOL copy storage pool every two days at 8 p.m.

To update the BACKUP_ARCHIVEPOOL schedule, enter:

```
update schedule backup_archivepool type=administrative
startime=20:00 period=3
```

Starting with today, the BACKUP_ARCHIVEPOOL schedule begins the backup every three days at 10 p.m.

Copying Schedules

You can create a new schedule by copying an existing administrative schedule. When you copy a schedule, Tivoli Storage Manager copies the following information:

- A description of the schedule
- All parameter values from the original schedule

You can then update the new schedule to meet your needs.

To copy the BACKUP_ARCHIVEPOOL administrative schedule and name the new schedule BCKSCHED, enter:

```
copy schedule backup_archivepool bcksched type=administrative
```

Deleting Schedules

To delete the administrative schedule ENGBKUP, enter:

```
delete schedule engbkup type=administrative
```

Managing Scheduled Event Records

Task	Required Privilege Class
Display information about events	Any administrator
Set the retention period for event records	System
Delete event records	System or unrestricted policy

Each scheduled administrative command operation is called an *event*. All scheduled events, including their status, are tracked by the server. An *event record* is created in the server database whenever processing of a scheduled command is created or missed.

Querying Events

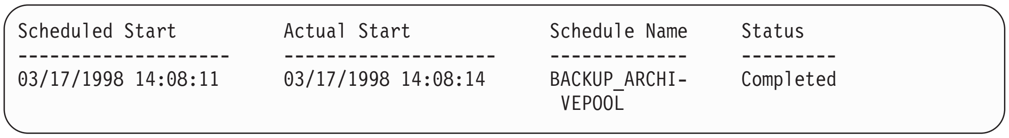
To help manage schedules for administrative commands, you can request information about scheduled and completed events. You can request general or exception reporting queries.

- To get information about past and projected scheduled processes, use a general query. If the time range you specify includes the future, the query output shows which events should occur in the future based on current schedules.
- To get information about scheduled processes that did not complete successfully, use exception reporting.

To minimize the processing time when querying events, minimize the time range.

To query an event for an administrative command schedule, you must specify the TYPE=ADMINISTRATIVE parameter. Figure 52 shows an example of the results of the following command:

```
query event * type=administrative
```



```
Scheduled Start          Actual Start             Schedule Name     Status
--------------------     --------------------     -------------     ---------
03/17/1998 14:08:11      03/17/1998 14:08:14      BACKUP_ARCHI-     Completed
                                                   VEPOOL
```

Figure 52. Query Results for an Administrative Schedule

Removing Event Records from the Database

You can specify how long event records stay in the database before the server deletes them. You can also manually remove event records from the database.

If you issue a query for events, past events may display even if the event records have been deleted. The events displayed with a status of *Uncertain* indicate that complete information is not available because the event records have been deleted. To determine if event records have been deleted, check the message that is issued after the DELETE EVENT command is processed.

Setting the Event Record Retention Period

You can specify the retention period for event records in the database. After the retention period passes, the server automatically removes the event records from the database. At installation, the retention period is set to 10 days.

To set the retention period to 15 days, enter:

```
set eventretention 15
```

Event records are automatically removed from the database after both of the following conditions are met:

- The specified retention period has passed
- The startup window for the event has elapsed

Deleting Event Records

Because event records are deleted automatically, you do not have to manually delete them from the database. However, you may want to manually delete event records to increase available database space.

To delete all event records written prior to 11:59 p.m. on June 30, 1998, enter:

```
delete event type=administrative 06/30/1998 23:59
```

Tivoli Storage Manager Server Scripts

Tivoli Storage Manager provides for automation of common administrative tasks with server scripts that are stored in the database. The scripts can be processed directly on the server console, the web interface, or included in an administrative command schedule. Tivoli Storage Manager provides sample scripts in *scripts.smp.* The sample scripts have an example order of execution for scheduling administrative commands. For more information, see “Using SELECT Commands in Tivoli Storage Manager Scripts” on page 409. The sample scripts can be loaded from the *scripts.smp* file by issuing the runfile command. See *Quick Start* for details.

The administrator can run the script by issuing the RUN command from the web administrative interface, or scheduling the script for processing using the administrative command scheduler on the server. If one of the specified commands in the script does not process successfully, the remaining commands are not processed.

Tivoli Storage Manager scripts can include the following:

- Command parameter substitution.
- SQL SELECT statements that you specify when the script is processed.
- Conditional logic flow statements. These logic flow statements include:
  - The IF clause; this clause determines how processing should proceed based on the current return code value.
  - The EXIT statement; this statement ends script processing.
  - The GOTO and LABEL statement; this statement directs logic flow to continue processing with the line that starts with the label specified.
  - Comment lines.

Defining a Server Script

Task	Required Privilege Class
Define a server script	System, policy, storage, and operator

You can define a server script line by line, create a file that contains the command lines, or copy an existing script.

The following examples use commands to define and update scripts. However, you can easily define and update scripts using the web administrative interface where you can also use local workstation cut and paste functions.

You can define a script with the DEFINE SCRIPT command. You can initially define the first line of the script with this command. For example:

```
define script qaixc "select node_name from nodes where platform='aix'"
desc='Display AIX clients'
```

This example defines the script as QAIXC. When you run the script, all AIX clients are displayed.

To define additional lines, use the UPDATE SCRIPT command. For example, you want to add a QUERY SESSION command, enter:

```
update script qaixc "query session *"
```

You can specify a WAIT parameter with the DEFINE CLIENTACTION command that allows the client action to complete before processing the next step in a command script or macro. See *Administrator's Reference* for information

You can use the ISSUE MESSAGE command to determine where a problem is within a command in a script. See *Administrator's Reference* for information on how to use the ISSUE MESSAGE command.

For additional information about updating server scripts, or updating a command line, see "Updating a Script" on page 376.

Defining a Server Script Using Contents of Another File

You can define a script whose command lines are read in from another file that contains statements for the script to be defined. For example, to define a script whose command lines are read in from the file BKUP12.MAC, issue:

```
define script admin1 file=bkup12.mac
```

The script is defined as ADMIN1, and the contents of the script have been read in from the file BKUP12.MAC.

Note: The file must reside on the server, and is read by the server.

Using Continuation Characters for Long Commands

You can continue long commands across multiple command lines by specifying the continuation character (-) as the last character for a command that is continued. The following example continues an SQL statement across multiple command lines:

```
/*-----------------------------*/
/* Sample continuation example */
SELECT-
* FROM-
NODE WHERE-
PLATFORM='win32'
```

When this command is processed, it runs the following:

```
select * from nodes where platform='win32'
```

Using Substitution Variables

You can include substitution variables in a script. Substitution variables are specified with a $ character followed by a number that represents the position of the parameter when the script is processed. The following example SQLSAMPLE script specifies substitution variables $1 and $2:

```
/*----------------------------------------------*/
/* Sample substitution example */
/* ---------------------------------------------*/
SELECT-
$1 FROM-
NODES WHERE-
PLATFORM='$2'
```

When you run the script you must specify two values, one for $1 and one for $2. For example:

```
run sqlsample node_name aix
```

The command that is processed when the SQLSAMPLE script is run is:

```
select node_name from nodes where platform='aix'
```

Using Logic Flow Statements in a Script

You can use conditional logic flow statements based on return codes issued from previous command processing. These logic statements allow you to process your scripts based on the outcome of certain commands. You can use IF, EXIT, or GOTO (label) statements.

As each command is processed in a script, the return code is saved for possible evaluation before the next command is processed. The return code can be one of three severities: OK, WARNING, or ERROR. Refer to *Administrator's Reference* for a list of valid return codes and severity levels.

Specifying the IF Clause: You can use the IF clause at the beginning of a command line to determine how processing of the script should proceed based on the current return code value. In the IF clause you specify a return code symbolic value or severity.

The server initially sets the return code at the beginning of the script to RC_OK. The return code is updated by each processed command. If the current return code from the processed command is equal to any of the return codes or severities in the IF clause, the remainder of the line is processed. If the current return code is not equal to one of the listed values, the line is skipped.

The following script example backs up the BACKUPPOOL storage pool only if there are no sessions currently accessing the server. The backup proceeds only if a return code of RC_NOTFOUND is received:

```
/* Backup storage pools if clients are not accessing the server */
select * from sessions
/* There are no sessions if rc_notfound is received */
if(rc_notfound) backup stg backuppool copypool
```

The following script example backs up the BACKUPPOOL storage pool if a return code with a severity of warning is encountered:

```
/* Backup storage pools if clients are not accessing the server */
select * from sessions
/* There are no sessions if rc_notfound is received */
if(warning) backup stg backuppool copypool
```

Specifying the EXIT Statement: The EXIT statement ends script processing. The following example uses the IF clause together with RC_OK to determine if clients are accessing the server. If a RC_OK return code is received, this indicates that client sessions are accessing the server. The script proceeds with the exit statement, and the backup does not start.

```
/* Back up storage pools if clients are not accessing the server */
select * from sessions
/* There are sessions if rc_ok is received */
if(rc_ok) exit
backup stg backuppool copypool
```

Specifying the GOTO Statement: The GOTO statement is used in conjunction with a label statement. The label statement is the target of the GOTO statement. The GOTO statement directs script processing to the line that contains the label statement to resume processing from that point. The label statement always has a colon (:) after it and may be blank after the colon.

The following example uses the GOTO statement to back up the storage pool only if there are no sessions currently accessing the server. In this example, the return code of RC_OK indicates that clients are accessing the server. The GOTO statement directs processing to the **done:** label which contains the EXIT statement that ends the script processing:

```
/* Back up storage pools if clients are not accessing the server */
select * from sessions
/* There are sessions if rc_ok is received */
if(rc_ok) goto done
backup stg backuppool copypool
done:exit
```

Managing Server Scripts

You can update, copy, rename, query, delete, and run server scripts.

Task	Required Privilege Class
Update, copy, rename, query, and delete a script Run a script	System, policy, storage, and operator

Updating a Script

You can update a script to change an existing command line or to add a new command line to a script.

To change an existing command line, specify the LINE= parameter.

To append a command line to an existing script issue the UPDATE SCRIPT command without the LINE= parameter. The appended command line is assigned a line number of five greater than the last command line number in the command line sequence. For example, if your script ends with line 010, the appended command line is assigned a line number of 015.

Appending a New Command: The following is an example of the QSTATUS script. The script has lines 001, 005, and 010 as follows:

```
001 /* This is the QSTATUS script */
005 QUERY STATUS
010 QUERY PROCESS
```

To append the QUERY SESSION command at the end of the script, issue the following:

```
update script qstatus "query session"
```

The QUERY SESSION command is assigned a command line number of 015 and the updated script is as follows:

```
001 /* This is the QSTATUS script */
005 QUERY STATUS
010 QUERY PROCESS
015 QUERY SESSION
```

Replacing an Existing Command: Line number 010 in the QSTATUS script contains a QUERY PROCESS command. To replace the QUERY PROCESS command with the QUERY STGPOOL command, specify the LINE= parameter as follows:

```
update script qstatus "query stgpool" line=10
```

The QSTATUS script is updated to the following:

```
001 /* This is the QSTATUS script */
005 QUERY STATUS
010 QUERY STGPOOL
015 QUERY SESSION
```

Adding a New Command and Line Number: To add the SET REGISTRATION OPEN command as the new line 007 in the QSTATUS script, issue the following:

```
update script qstatus "set registration open" line=7
```

The QSTATUS script is updated to the following:

```
001 /* This is the QSTATUS script */
005 QUERY STATUS
007 SET REGISTRATION OPEN
010 QUERY STGPOOL
015 QUERY SESSION
```

Copying a Server Script

You can copy an existing script to a new script with a different name. For example, to copy the QSTATUS script to QUERY1 script, issue:

```
copy script qstatus query1
```

The QUERY1 command script now contains the same command lines as the QSTATUS command script.

Querying a Server Script

You can query a script to display information about the script. You can specify wildcard characters to display all scripts with names that match a particular pattern. When you query a script, you can direct the output to a file in a file system that the server can access. The various formats you can use to query scripts are as follows:

Format	Description
Standard	Displays the script name and description. This is the default.
Detailed	Displays commands in the script and their line numbers, date of last update, and update administrator for each command line in the script.
Lines	Displays the name of the script, the line numbers of the commands, comment lines, and the commands.
Raw	Outputs only the commands contained in the script without all other attributes. You can use this format to direct the script to a file so that it can be loaded into another server with the DEFINE script command specifying the FILE= parameter.

The following is an example for querying a script in the standard format.

```
query script *
```

The command gives results like the following:

```
Name            Description
--------------- ------------------------------------------------------
QCOLS           Display columns for a specified SQL table
QSAMPLE         Sample SQL Query
```

For more information about querying a server script, refer to *Administrator's Reference*.

Querying a Server Script to Create Another Server Script: You can create additional server scripts by querying a script and specifying the FORMAT=RAW and OUTPUTFILE parameters. You can use the resulting output as input into another script without having to create a script line by line.

The following is an example of querying the SRTL2 script in the raw format, directing the output to newscript.script:

```
query script srtl2 format=raw outputfile=newscript.script
```

You can then edit the newscript.script with an editor that is available to you on your system. To create a new script using the edited output from your query, issue:

```
define script srtnew file=newscript.script
```

Renaming a Server Script

You can rename a script to a different name. For example, to rename the QUERY1 script to QUERY5, issue:

```
rename script query1 query5
```

The QUERY1 script is now named QUERY5.

Deleting a Command from a Server Script

You can delete an individual command line from a script. When you specify a line number, only the corresponding command line is deleted from the script.

For example, to delete the 007 command line from the QSTATUS script, issue:

```
delete script qstatus line=7
```

Deleting a Server Script

To delete an entire script, issue the DELETE SCRIPT command.

To delete the QSTATUS script, issue:

```
delete script qstatus
```

Running a Server Script

To process a script, issue the RUN command. You can run a script that contains substitution variables by specifying them along with the RUN command.

You can preview the command lines of a script without actually executing the commands by using the PREVIEW=YES parameter with the RUN command. If the script contains substitution variables, the command lines are displayed with the substituted variables. This is useful for evaluating a script before you run it.

For example, to process the QAIXC script previously defined, issue:

```
run qaixc
```

To process the following script that contains substitution variables:

```
/*----------------------------------------------*/
/* Sample continuation and substitution example */
/* ---------------------------------------------*/
SELECT-
$1 FROM-
NODES WHERE-
PLATFORM='$2'
```

Enter:

```
run qaixc node_name aix
```

Where $1 is node_name and $2 is aix.

Using Macros

Tivoli Storage Manager supports macros on the administrative client. A macro is a file that contains one or more administrative client commands. You can only run a macro from the administrative client in batch or interactive modes. Macros are stored as a file on the administrative client. Macros are not distributed across servers and cannot be scheduled on the server.

Macros can include the following:

- Administrative commands

  For more information on administrative commands, see "Writing Commands in a Macro".
- Comments

  For more information on comments, see "Writing Comments in a Macro" on page 380.
- Continuation characters

  For more information on continuation characters, see "Using Continuation Characters" on page 380.
- Variables

  For more information on variables, see "Using Substitution Variables in a Macro" on page 381.

The name for a macro must follow the naming conventions of the administrative client running on your operating system. For more information about file naming conventions, refer to the *Administrator's Reference*.

In macros that contain several commands, use the COMMIT and ROLLBACK commands to control command processing within the macro. For more information about using these commands, see "Controlling Command Processing in a Macro" on page 382.

You can include the MACRO command within a macro file to invoke other macros up to ten levels deep. A macro invoked from the Tivoli Storage Manager administrative client command prompt is called a high-level macro. Any macros invoked from within the high-level macro are called *nested* macros.

Writing Commands in a Macro

Refer to the *Administrator's Reference* for more information on how commands are entered and the general rules for entering administrative commands. The administrative client ignores any blank lines included in your macro. However, a completely blank line terminates a command that is continued (with a continuation character).

The following is an example of a macro called REG.MAC that registers and grants authority to a new administrator:

```
register admin pease mypasswd -
  contact='david pease, x1234'
grant authority pease -
  classes=policy,storage -
  domains=domain1,domain2 -
  stgpools=stgpool1,stgpool2
```

This example uses continuation characters in the macro file. For more information on continuation characters, see "Using Continuation Characters" on page 380.

After you create a macro file, you can update the information that it contains and use it again. You can also copy the macro file, make changes to the copy, and then run the copy.

Writing Comments in a Macro

You can add comments to your macro file. To write a comment, write a slash and an asterisk (/*) to indicate the beginning of the comment, write the comment, and then write an asterisk and a slash (*/) to indicate the end of the comment. You can put a comment on a line by itself, or you can put it on a line that contains a command or part of a command.

For example, to use a comment to identify the purpose of a macro, write the following:

```
/* auth.mac-register new nodes  */
```

Or, to write a comment to explain something about a command or part of a command, write:

```
domain=domain1           /*assign node to domain1  */
```

Comments cannot be nested and cannot span lines. Every line of a comment must contain the comment delimiters.

Using Continuation Characters

You can use continuation characters in a macro file. Continuation characters are useful when you want to execute a command that is longer than your screen or window width.

Attention: Without continuation characters, you can enter up to 256 characters. With continuation characters, you can enter up to 1500 characters. In the MACRO command, these maximums are *after* any substitution variables have been applied (see "Using Substitution Variables in a Macro" on page 381).

To use a continuation character, enter a dash or a back slash at the end of the line that you want to continue. With continuation characters, you can do the following:

- Continue a command. For example:

  ```
  register admin pease mypasswd -
  contact="david, ext1234"
  ```

- Continue a list of values by entering a dash or a back slash, with no preceding blank spaces, after the last comma of the list that you enter on the first line. Then, enter the remaining items in the list on the next line with no preceding blank spaces. For example:

  ```
  stgpools=stg1,stg2,stg3,-
  stg4,stg5,stg6
  ```

- Continue a string of values enclosed in quotation marks by entering the first part of the string enclosed in quotation marks, followed by a dash or a back slash at the end of the line. Then, enter the remainder of the string on the next line enclosed in the *same* type of quotation marks. For example:

  ```
  contact="david pease, bldg. 100, room 2b, san jose,"-
  "ext. 1234, alternate contact-norm pass,ext 2345"
  ```

  Tivoli Storage Manager concatenates the two strings with no intervening blanks. You must use *only* this method to continue a quoted string of values across more than one line.

Using Substitution Variables in a Macro

You can use substitution variables in a macro to supply values for commands when you run the macro. When you use substitution variables, you can use a macro again and again, whenever you need to perform the same task for different objects or with different parameter values.

A substitution variable consists of a percent sign (%), followed by a number that indicates the number of the substitution variable. When you run the file with the MACRO command, you must specify values for the variables.

For example, to create a macro named AUTH.MAC to register new nodes, write it as follows:

```
/* register new nodes */
register node %1 %2 -      /*  userid password                  */
  contact=%3 -             /* 'name, phone number'              */
  domain=%4                /*  policy domain                    */
```

Then, when you run the macro, you enter the values you want to pass to the server to process the command.

For example, to register the node named DAVID with a password of DAVIDPW, with his name and phone number included as contact information, and assign him to the DOMAIN1 policy domain, enter:

```
macro auth.mac david davidpw "david pease, x1234" domain1
```

If your system uses the percent sign as a wildcard character, a pattern-matching expression in a macro where the percent sign is immediately followed by a numeric digit is interpreted by the administrative client as a substitution variable.

You cannot enclose a substitution variable in quotation marks. However, a value you supply as a substitution for the variable can be a quoted string.

Running a Macro

Use the MACRO command when you want to run a macro. You can enter the MACRO command in batch or interactive mode.

If the macro does not contain substitution variables (such as the REG.MAC macro described in the “Writing Commands in a Macro” on page 379), run the macro by entering the MACRO command with the name of the macro file. For example:

```
macro reg.mac
```

If the macro contains substitution variables (such as the AUTH.MAC macro described in “Using Substitution Variables in a Macro”), include the values that you want to supply after the name of the macro. Each value is delimited by a space. For example:

```
macro auth.mac pease mypasswd "david pease, x1234" domain1
```

If you enter fewer values than there are substitution variables in the macro, the administrative client replaces the remaining variables with null strings.

If you want to omit one or more values between values, enter a null string ("") for each omitted value.

For example, if you omit the contact information in the previous example, you must enter:

```
macro auth.mac pease mypasswd "" domain1
```

Controlling Command Processing in a Macro

When you issue a MACRO command, the server processes all commands in the macro file in order, including commands contained in any nested macros. The server commits all commands in a macro after successfully completing processing for the highest-level macro. If an error occurs in any command in the macro or in any nested macro, the server terminates processing and rolls back any changes caused by all previous commands.

If you specify the ITEMCOMMIT option when you enter the DSMADMC command, the server commits each command in a script or a macro individually, after successfully completing processing for each command. If an error occurs, the server continues processing and only rolls back changes caused by the failed command.

You can control precisely when commands are committed with the COMMIT command. If an error occurs while processing the commands in a macro, the server terminates processing of the macro and rolls back any uncommitted changes (commands that have been processed since the last COMMIT). Make sure that your administrative client session is *not* running with the ITEMCOMMIT option if you want to control command processing with the COMMIT command.

Note: Commands that start background processes cannot be rolled back. For a list of commands that can generate background processes, see "Managing Server Processes" on page 360.

You can test a macro before implementing it by using the ROLLBACK command. You can enter the commands (except the COMMIT command) you want to issue in the macro, and enter ROLLBACK as the last command. Then, you can run the macro to verify that all the commands process successfully. Any changes to the database caused by the commands are rolled back by the ROLLBACK command you have included at the end. Remember to remove the ROLLBACK command before you make the macro available for actual use. Also, make sure your administrative client session is not running with the ITEMCOMMIT option if you want to control command processing with the ROLLBACK command.

If you have a series of commands that process successfully via the command line, but are unsuccessful when issued within a macro, there are probably dependencies between commands. It is possible that a command issued within a macro cannot be processed successfully until a previous command that is issued within the same macro is committed. Either of the following actions allow successful processing of these commands within a macro:

- Insert a COMMIT command before the command dependent on a previous command. For example, if COMMAND C is dependent upon COMMAND B, you would insert a COMMIT command before COMMAND C. An example of this macro is:

  ```
  command a
  command b
  commit
  command c/
  ```

- Start the administrative client session using the ITEMCOMMIT option. This causes each command within a macro to be committed before the next command is processed.

Chapter 18. Managing the Database and Recovery Log

The Tivoli Storage Manager database contains information that is needed for server operations and information about client data that has been backed up, archived, and space-managed. The database does not store client data. Instead, the database points to the locations of the client files in the storage pools.

The database includes information about:

- Client nodes and administrators
- Policies and schedules
- Server settings
- Locations of client files on server storage
- Server operations (for example, activity logs and event records)

Note: If the database is unusable, the entire Tivoli Storage Manager server is unavailable. If a database is lost and cannot be recovered, the backup, archive, and space-managed data for that server is lost. See "Chapter 22. Protecting and Recovering Your Server" on page 499 for steps that you can take to protect your database.

The recovery log contains information about database updates that have not yet been committed. Updates can include activities such as defining a management class, backing up a client file, and registering a client node. Changes to the database are recorded in the recovery log to maintain a consistent database image.

The following shows authority requirements for tasks in this chapter:

Task	**Required Privilege Class**
Manage disk volumes used by the database and recovery log	System or unrestricted storage
Display information about the database and recovery log	Any administrator

See the following sections:

Concepts:
"How Tivoli Storage Manager Processes Transactions" on page 384
"How Tivoli Storage Manager Manages Space" on page 384
"The Advantages of Using Journal File System Files" on page 385

Tasks:
"Estimating and Monitoring Database and Recovery Log Space Requirements" on page 386
"Increasing the Size of the Database or Recovery Log" on page 388
"Decreasing the Size of the Database or Recovery Log" on page 392
"Optimizing Database and Recovery Log Performance" on page 395

© Copyright IBM Corp. 1993, 2002

Note: Mirroring of the database and recovery log is described in the chapter on data protection. See "Mirroring the Database and Recovery Log" on page 504.

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

How Tivoli Storage Manager Processes Transactions

To support multiple transactions from concurrent client sessions, the server holds transaction log records in the recovery log buffer pool until they can be written to the recovery log. These records remain in the buffer pool until the active buffer becomes full or Tivoli Storage Manager forces log records to the recovery log.

Changes resulting from transactions are held in the buffer pool temporarily and are not made to the database immediately. Therefore, the database and recovery log are not always consistent. When all records for a transaction are written to the recovery log, Tivoli Storage Manager updates the database. The transaction is then committed to the database. At some point after a transaction is committed, the server deletes the transaction record from the recovery log.

How Tivoli Storage Manager Manages Space

Tivoli Storage Manager tracks all volumes defined to the database as one logical volume and all volumes defined to the recovery log as another logical volume. In Figure 53, the database consists of four volumes: VOL1 through VOL4, which the server tracks as a single logical volume.

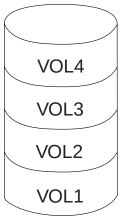


Figure 53. A Server Database

To manage the database and recovery log effectively, you must understand the following concepts:

- Available space
- Assigned capacity
- Utilization

Available Space

Not all of the space that is allocated for the volumes of the database or of the recovery log can be used for database and recovery log information. The server subtracts 1MB from each physical volume for overhead. The remaining space is divided into 4MB partitions. For example, you allocate four 25MB volumes for the database. For the four volumes, 4MB are needed for overhead leaving 96MB of available space as shown in figure Figure 54:

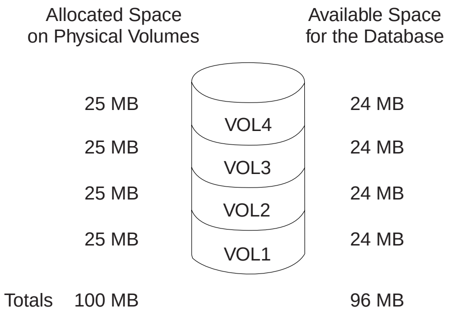


Figure 54. An Example of Available Space

Assigned Capacity

Assigned capacity is the available space that can be used for database or recovery log information. During installation, the assigned capacities of the database and recovery log match the available space. If you add volumes after installation, you increase your available space. However, to increase the assigned capacity, you must also extend the database or recovery log. See “Step 2: Extending the Capacity of the Database or Recovery Log” on page 392 for details.

Utilization

Utilization is the percent of the assigned capacity in use at a specific time. *Maximum percent utilized* is the highest utilization since the statistics were reset. For example, an installation performs most backups after midnight. Figure 55 shows that utilization statistics for the recovery log were reset at 9 p.m. the previous evening and that the maximum utilization occurred at 12 a.m.

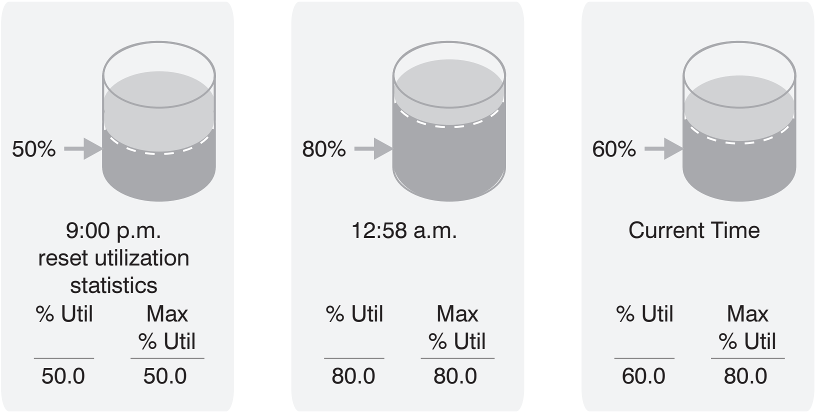


Figure 55. An Example of Recovery Log Utilization

Unless many objects are deleted, the database maximum percent utilized is usually close to the utilization percentage.

The Advantages of Using Journal File System Files

Tivoli Storage Manager supports both journaled file system (JFS) files and raw logical volumes as database, recovery log, and disk storage pool volumes. JFS files have the following advantages:

- JFS locks open files, and other applications cannot write to them. However, raw volumes are not locked, and any application can write to them. Tivoli Storage Manager tries to prevent starting more than one instance of the same server from the same directory, but it can be done. If you are using raw volumes, both server instances can simultaneously update the same information. This could cause errors in the database, recovery log, or storage pool raw volumes.
- After a database, recovery log, or storage pool volume is defined, you cannot change its size. Size information determines where data is placed and whether volumes have been modified by other applications or utilities. However, smit lets you increase the sizes of raw volume. If the volume is defined before its size is increased, Tivoli Storage Manager cannot use the volume or its data.
- For database and recovery log volumes you should use Tivoli Storage Manager mirroring rather than AIX mirroring. If you use AIX mirroring for database and recovery log volumes, you may have a problem with raw volumes, but not with JFS files. AIX tracks mirroring activity by writing control information to the first 512 bytes of USER area in a raw volume. This is not a problem for database and recovery log volumes, but Tivoli Storage Manager control information is also written in this area. If AIX overwrites this control information when raw volumes are mirrored, Tivoli Storage Manager may not be able to vary the volume online.

  Note: Tivoli Storage Manager mirroring only supports database and recovery log volumes. Disk storage pool volumes are not supported by Tivoli Storage Manager mirroring.

The use of JFS files for database, recovery log, and storage pool volumes requires slightly more CPU than is required for raw volumes. However, JFS read-ahead caching improves performance.

Estimating and Monitoring Database and Recovery Log Space Requirements

The size of the database depends on the number of client files to be stored and the method by which the server manages them. If you can estimate the maximum number of files that might be in server storage at any time, you can estimate the database size from the following information:

- Each stored **version of a file** requires about 400 to 600 bytes of database space.
- Each **cached** or **copy storage pool** file requires about 100 to 200 bytes of database space.
- **Overhead** could require up to 25% in additional space.

In the example below, the computations are probable maximums. In addition, the numbers are not based on the use of file aggregation. In general, aggregation of small files reduces the required database space. For details about aggregation, see “How the Server Groups Files before Storing” on page 171. Assume the following numbers for a Tivoli Storage Manager system:

Versions of files

Backed up files

Up to 500 000 client files might be backed up. Storage policies call for keeping up to 3 copies of backed up files:

```
500 000 files x 3 copies = 1 500 000 files
```

Archived files

Up to 100 000 files might be archived copies of client files.

Space-managed files

Up to 200 000 files migrated from client workstations might be in server storage.

Note: File aggregation does not affect space-managed files.

At 600 bytes per file, the space required for these files is:

```
(1 500 000 + 100 000 + 200 000) x 600 = 1.0GB
```

Cached and copy storage pool files

Cached copies

Caching is enabled in a 5GB disk storage pool. The pool's high and low migration thresholds are 90% and 70% respectively. Thus, 20% of the disk pool, or 1GB, is occupied by cached files.

If the average file size is about 10KB, about 100 000 files are in cache at any one time.

```
100 000 files x 200 bytes = 19MB
```

Copy storage pool files

All primary storage pools are backed up to the copy storage pool:

```
(1 500 000 + 100 000 + 200 000) x 200 bytes = 343MB
```

Therefore, cached files and copy storage pool files require about 0.4GB of database space.

Overhead

About 1.4GB is required for file versions and cached and copy storage pool files. Up to 50% additional space (or 0.7GB) should be allowed for overhead.

The database should then be approximately 2.1GB.

If you cannot estimate the numbers of files, you can roughly estimate the database size as from 1% to 5% of the required server storage space. For example, if you need 100GB of server storage, your database should be between 1GB and 5GB. See "Estimating Space Needs for Storage Pools" on page 196 for details.

During SQL queries of the server, intermediate results are stored in temporary tables that require space in the free portion of the database. Therefore, the use of SQL queries requires additional database space. The more complicated the queries, the greater is the space required.

The size of the recovery log depends on the number of concurrent client sessions and the number of background processes executing on the server. The maximum number of concurrent client sessions is set in the server options.

Attention: Be aware that the results are estimates. The actual size of the database may differ from the estimate because of factors such as the number of directories and the length of the path and file names. You should periodically monitor your database and recovery log and adjust their sizes as necessary.

Begin with at least a 12MB recovery log. If you use the database backup and recovery functions in roll-forward mode, you should begin with at least 25MB. See "Database and Recovery Log Protection" on page 502 and "Estimating the Size of the Recovery Log" on page 511 for more information.

Monitoring the Database and Recovery Log

You should regularly monitor the database and recovery log to see if you should add or delete space. To monitor daily utilization, you might want to reset the maximum utilization counters each day. Utilization statistics are reset in two ways:

- Automatically when the server is restarted
- By issuing the RESET DBMAXUTILIZATION or RESET LOGMAXUTILIZATION commands

For example, to reset the database utilization statistic, enter:

```
reset dbmaxutilization
```

If the SELFTUNEBUFPOOLSIZE server option is in effect, the buffer pool cache hit ratio statistics are reset at the start of expiration. After expiration, the buffer pool size is increased if the cache hit ratio is less than 98%. The increase in the buffer pool size is in small increments and may change after each expiration. The change in the buffer pool size is not reflected in the server options file. You can check the current size at any time using the QUERY STATUS command. Use the SETOPT BUFPOOLSIZE command to change the buffer pool size.

To display information about the database or recovery log, issue the QUERY DB or QUERY LOG. For example:

```
query db
```

The server displays a report, like this:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
       96       96         0        92   4,096    24,576        86   0.3   0.3
```

See the indicated sections for details about the following entries:

- Available space, "Available Space" on page 384
- Assigned capacity, "Assigned Capacity" on page 385
- Utilization and maximum utilization, "Utilization" on page 385

If utilization is high, you may want to add space (see "Increasing the Size of the Database or Recovery Log"). If utilization is low, you may want to delete space (see "Decreasing the Size of the Database or Recovery Log" on page 392).

Note: You can also use a DEFINE SPACETRIGGER command to automatically check whether the database or recovery log exceeds a utilization percentage that you specify. See "Automating the Increase of the Database or Recovery Log" on page 389 for details.

Increasing the Size of the Database or Recovery Log

As your requirements change, you can increase or decrease the sizes of the database and recovery log. You can automate the process of increasing the sizes, or you can perform all the steps manually. See "Automating the Increase of the Database or Recovery Log" on page 389 or "Manually Increasing the Database or Recovery Log" on page 390.

Attention: Do not change the size of an allocated database or recovery log volume after it has been defined. If you change the size of a volume, Tivoli Storage Manager may not initialize correctly, and data may be lost.

Note: Significantly increasing the recovery log size could significantly increase the time required to restart the server, back up the database, and restore the database.

Automating the Increase of the Database or Recovery Log

You can automate the process of increasing the database and recovery log sizes. With a DEFINE SPACETRIGGER command, you can specify the following:

- Utilization percentages at which the database or recovery log size is to be increased
- The size of the increase as a percentage of the current database or recovery log size
- The prefix to be used for a new volume
- The maximum size allowed for the database or recovery log

For example, assume that you have a 100GB database and a 3GB recovery log. You want to increase the database size by 25 percent when 85 percent is in use, but not to more than 200GB. You also want to increase the recovery log size by 30 percent when 75 percent is in use, but not to more than 5GB.

Note: There is one time when the database or recovery log might exceed the maximum size specified: If the database or recovery log is less than the maximum size when expansion begins, it continues to the full expansion value. However, no further expansion will occur unless the space trigger is updated.

To add the new volumes to the */usr/lpp/adsmserv/bin/* directory, issue the following commands:

```
define spacetrigger db fullpct=85 spaceexpansion=25
expansionprefix=/usr/lpp/adsmserv/bin/ maximumsize=200000

define spacetrigger log fullpct=75 spaceexpansion=30
expansionprefix=/usr/lpp/adsmserv/bin/ maximumsize=50000
```

The server then monitors the database or recovery log and, if the utilization level is reached, does the following:

- Displays a message (ANR4413I or ANR4414I) that states the amount of space required to meet the utilization parameter specified in the command.
- Allocates space for the new volume.
- Defines the new volume.
- Extends the database or recovery log.
- If a volume is mirrored and there is enough disk space, the preceding steps are also performed for the mirrored copies.

Notes:

1. The maximum size of the recovery log is 13GB. The server will not automatically extend the recovery log beyond 12GB.
2. An automatic expansion may exceed the specified database or recovery log maximum size but not the 13GB recovery log limit. However, after the maximum has been reached, no further automatic expansions will occur.

3. A space trigger percentage may be exceeded between the monitoring of the database or recovery log and the time that a new volume is brought online.
4. If the server creates a database or recovery log volume and the attempt to add it to the server fails, the volume is not deleted. After the problem is corrected, you can define it with the DEFINE DBVOLUME or DEFINE LOGVOLUME command.
5. Automatic expansion will not occur during a database backup.
6. The database and recovery log utilization percentage may not always be below the space trigger value. The server checks utilization after a database or recovery log commit.

   Also, deleting database volumes and reducing the database does not activate the trigger. Therefore, the utilization percentage can exceed the set value before new volumes are online.
7. Setting a maximum size does not mean that the database and recovery log will always be less than that value. The value is a threshold for expansion. The server does not automatically expand the database or recovery log if its size is greater than the maximum size. The server checks the size and allows expansion if the database or recovery log is less than the maximum size. The server only checks the size that results after expansion to ensure that maximum recovery log size is not exceeded.

Recovering When the Recovery Log Runs Out of Space

If the log mode is set to ROLLFORWARD and either the recovery log is too small or the database backup trigger is set too high, the recovery log could run out of space before database operations complete. If this happens, you may need to stop the server without enough recovery log space to restart the server. In some cases, the server halts itself.

To restart the server, first format a new volume (see “Using the DSMFMT Command to Format Volumes” on page 392). Then use the DSMSERV EXTEND LOG command to extend the size of the recovery log. For example, after formatting a 21MB volume named *new.reclog*, extend the recovery log by issuing the following command:

```
dsmserv extend log new.reclog 20
```

After the server is running, you can do the following:

- Back up the database, which frees the recovery log space
- Adjust the size of the recovery log, the database backup trigger, or both

Manually Increasing the Database or Recovery Log

To add space to the database or recovery log, do the following:

“Step 1: Creating Database and Recovery Log Volumes”
“Step 2: Extending the Capacity of the Database or Recovery Log” on page 392

Step 1: Creating Database and Recovery Log Volumes

You can allocate space and define a database or recovery log volume in a single operation. For example, to allocate a 100MB database volume named VOL5 in the /usr/lpp/adsmserv/bin directory and define the volume, enter:

```
define dbvolume /usr/lpp/adsmserv/bin/vol5 formatsize=100
```

The available space of the database increases to 196MB, but the assigned capacity remains at 96MB. For Tivoli Storage Manager to use the space, you must extend the capacity (see “Step 2: Extending the Capacity of the Database or Recovery Log” on page 392). To verify the change, query the database or recovery log. For example, to query the database, enter:

```
query db
```

The server displays a report, like this:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
      196       96       100        92   4,096    24,576        86   0.3   0.3
```

The value in the *Maximum Extension* field should equal the available space of the new volume. In this example, a 101MB volume was allocated. This report shows that the available space has increased by 100MB; the assigned capacity is unchanged at 96MB; and the maximum extension is 100MB. Figure 56 illustrates these changes.

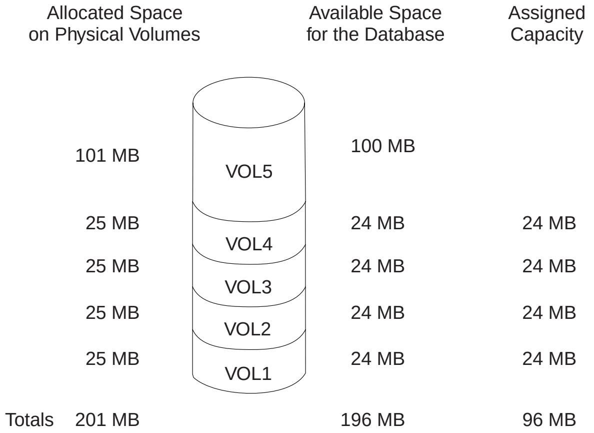


Figure 56. Adding Volumes Increases Available Space

You can also query the database and recovery log volumes to display information about the physical volumes that make up the database and recovery log.

Notes:

1. The maximum size of the recovery log is 13GB, and the maximum size of the database is 530GB. If you allocate a volume that would cause the recovery log or database to exceed these limits, the subsequent DEFINE DBVOLUME or DEFINE LOGVOLUME command for the volume will fail.
2. For performance reasons, define more than one volume for the database and recovery log, and put these volumes on separate disks. This allows simultaneous access to different parts of the database or recovery log.
3. To use disk space efficiently, allocate a few large disk volumes rather than many small disk volumes. In this way, you avoid losing space to overhead processing.

   If you already have a number of small volumes and want to consolidate the space into one large volume, see “Decreasing the Size of the Database or Recovery Log” on page 392.
4. To protect database and recovery log volumes from media failure, use mirroring. See “Mirroring the Database and Recovery Log” on page 504 for details.

Using the DSMFMT Command to Format Volumes: You can still use the DSMFMT utility to allocate a database or recovery log volume. You would then issue the DEFINE DBVOLUME or DEFINE LOGVOLUME command without the FORMATSIZE parameter, and extend the database or recovery log (see “Step 2: Extending the Capacity of the Database or Recovery Log”).

To allocate an additional 101MB to the database as volume VOL5, enter:

```
> dsmfmt -db vol5 101
```

Step 2: Extending the Capacity of the Database or Recovery Log

The database and recovery log are extended in 4MB increments. If you do not specify the extension in 4MB increments, the server rounds up to the next 4MB partition. For example, if you specify 1MB, the server extends the capacity by 4MB.

To increase the capacity of the database by 100MB, enter:

```
extend db 100
```

After the database has been extended, the available space and assigned capacity are both equal to 196MB, as shown in Figure 57.

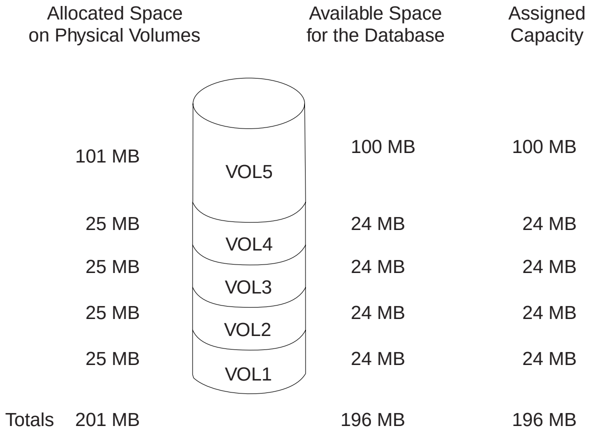


Figure 57. Extending the Capacity of the Database

You can query the database or recovery log (QUERY DB and QUERY LOG commands) to verify their assigned capacities. The server would display a report, like this:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
      196      196         0       192   4,096    50,176       111   0.2   0.2
```

Decreasing the Size of the Database or Recovery Log

You may want to delete database or recovery log volumes for a number of reasons. For example:

- You have a significant amount of space that is unused.
- You want to consolidate a number of small volumes, each of which may have unusable space, into one large volume. To create a volume, see “Increasing the Size of the Database or Recovery Log” on page 388.

When you delete a database or recovery log volume, Tivoli Storage Manager tries to move data from the volume being deleted to other physical volumes in the database or recovery log.

To delete space, perform the following steps:

1. Determine if you can delete one or more volumes ("Step 1: Determining If Volumes Can Be Deleted").
2. Reduce the capacity of the database or recovery log to free existing space ("Step 2: Reducing the Capacity of the Database or Recovery Log" on page 394).
3. Delete the volume ("Step 3: Deleting a Volume from the Database or Recovery Log" on page 394).

Step 1: Determining If Volumes Can Be Deleted

To determine if volumes can be deleted from the database or recovery log, check the volume sizes and the amount of unused space.

To check the sizes of the volumes in the database, enter:

```
query dbvolume format=detailed
```

The server displays the following type of information:

```
Volume Name (Copy 1): VOL1
         Copy Status: Sync'd
Volume Name (Copy 2):
         Copy Status: Undefined
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 24
Allocated Space (MB): 24
     Free Space (MB): 0
```

In this example, VOL1, VOL2, VOL3, and VOL4 each have 24MB of available space, and VOL5 has 100MB. To determine if there is enough unused space to delete one or more volumes, enter:

```
query db
```

The server displays the following type of report.

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
      196      196         0       176   4,096    50,176     4,755   9.5   9.5
```

The *Maximum Reduction* field shows the assigned capacity not in use. In this example, you could reduce the database by up to 176MB. This is enough space to allow the deletion of VOL1, VOL2, VOL3, and VOL4.

If there is not enough space on the remaining volumes, allocate more space and define an additional volume, as described in "Increasing the Size of the Database or Recovery Log" on page 388 and continue with "Step 2: Reducing the Capacity of the Database or Recovery Log" on page 394.

Step 2: Reducing the Capacity of the Database or Recovery Log

The database or recovery log capacity is reduced in 4MB increments. For example, based on the utilization of the database assume that VOL5 alone could contain all the data. To reduce the database by the amount of available space in VOL1 through VOL4, 96MB, enter:

```
reduce db 96
```

Reducing capacity is run as a background process and can take a long time. Issue a QUERY PROCESS command to check on the status of the process.

After reducing the database by 96MB, the assigned capacity is 100MB, and the maximum extension is 96MB, as shown in the following example:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
      196      100        96        92   4,096    24,576        86   0.3   0.3
```

Step 3: Deleting a Volume from the Database or Recovery Log

After you reduce the database or recovery log, use the smaller size for a few days. If the maximum utilization does not exceed 70%, you can delete the extra volumes.

Note: You cannot delete volumes if there is not enough free space for the server to move data from the volume being deleted to other physical volumes in the database or recovery log.

In our example, you determined that you can delete the four 24MB volumes from the database. You have reduced the database by 96MB. To delete VOL1 through VOL4 from the database, enter:

```
delete dbvolume vol1
delete dbvolume vol2
delete dbvolume vol3
delete dbvolume vol4
```

The server moves data from the volumes being deleted to available space on other volumes, as shown in Figure 58 on page 395.

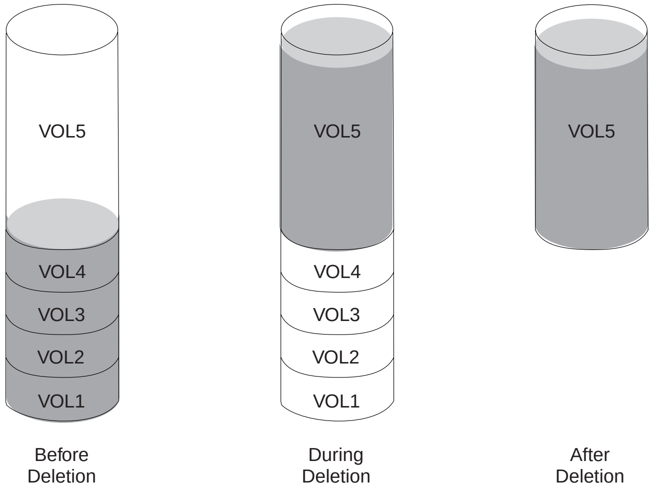


Figure 58. Deleting Database Volumes

After the data has been moved, these volumes are deleted from the server.

Optimizing Database and Recovery Log Performance

Over time, the database size and organization can change to the point that performance is degraded. Unloading and reloading the database can have the following benefits:

- Improved performance of the server database dump and load functions
- Improved performance of the database audit functions
- Improved use of database space
- Reorganization of fragmented page allocations
- Improved performance of long-running scans of the database

The database and recovery log buffer pool sizes can also affect performance. A larger database buffer pool can improve performance. A larger recovery log buffer pool reduces how often the server forces records to the recovery log.

See “Reorganizing the Database” on page 397 for more information about restoring database efficiency.

Adjusting the Database Buffer Pool Size

You can let Tivoli Storage Manager dynamically adjust the size of the database buffer pool or you can adjust it manually. If you specify YES for the SELFTUNEBUFPOOLSIZE server option, the database buffer pool is dynamically adjusted. The cache hit ratio statistics for the buffer pool are reset at the beginning of expiration. After expiration processing completes, the buffer pool size is adjusted dynamically.

Server expiration processing resets the database buffer pool before the next processing starts and examines if the database buffer pool cache hit ratio is above 98%. If the cache hit ratio is lower than 98%, the database buffer pool will be increased; if it is higher, the buffer pool size will not change. Increasing the database buffer pool will not be more than 10% of available real storage.

Manually Adjusting the Database Buffer Pool Size

Perform the following steps to track the database buffer pool statistics and adjust the buffer pool size:

Step 1: Reset Database Buffer Pool Utilization Statistics

Reset the buffer pool statistics. Initially, you might want to reset the statistics twice a day. Later, you can reset them less often. To reset, enter:

```
reset bufpool
```

Step 2: Monitor the Database Buffer Pool

To see if the database buffer pool is adequate for database performance, enter:

```
query db format=detailed
```

The server displays a report, like this:

```
  Available Space (MB): 196
Assigned Capacity (MB): 196
Maximum Extension (MB): 0
Maximum Reduction (MB): 176
     Page Size (bytes): 4,096
           Total Pages: 50,176
            Used Pages: 4,755
                 %Util: 9.5
            Max. %Util: 9.5
      Physical Volumes: 5
     Buffer Pool Pages: 128
 Total Buffer Requests: 1,193,212
        Cache Hit Pct.: 99.73
       Cache Wait Pct.: 0.00
```

Use the following fields to evaluate your current use of the database buffer pool:

Buffer Pool Pages
The number of pages in the database buffer pool. This value is determined by the server option for the size of the database buffer pool. At installation, the database buffer pool is set to 2048KB, which equals 128 database pages.

Total Buffer Requests
The number of requests for database pages since the server was last started or the buffer pool was last reset. If you regularly reset the buffer pool, you can see trends over time.

Cache Hit Pct
The percentage of requests for cached database pages in the database buffer pool that were not read from disk. A high value indicates that the size of your database buffer pool is adequate. If the value falls below 98%, consider increasing the size of the database buffer pool. For larger installations, performance could improve significantly if your cache hit percentage is greater than 99%.

Cache Wait Pct
The percentage of requests for database pages that had to wait for a buffer to become available in the database buffer pool. When this value is greater than 0, increase the size of the database buffer pool.

Step 3: Adjust the Database Buffer Pool

Use the BUFPOOLSIZE server option to set the size of the database buffer pool.

Adjusting the Recovery Log Buffer Pool Size

Do the following to adjust the size of the recovery log buffer pool:

Step 1: Monitor the Recovery Log Buffer Pool

To see how the recovery log buffer pool size affects recovery log performance, enter:

```
query log format=detailed
```

The server displays a report, like this:

```
  Available Space (MB): 12
Assigned Capacity (MB): 12
Maximum Extension (MB): 0
Maximum Reduction (MB): 8
     Page Size (bytes): 4,096
           Total Pages: 3,072
            Used Pages: 227
                 %Util: 7.4
            Max. %Util: 69.6
      Physical Volumes: 1
        Log Pool Pages: 32
    Log Pool Pct. Util: 6.25
    Log Pool Pct. Wait: 0.00
```

Use the following fields to evaluate the log buffer pool size:

Log Pool Pages

The number of pages in the recovery log buffer pool. This value is set by the server option for the size of the recovery log buffer pool. At installation, the default setting is 128KB, which equals 32 recovery log pages.

Log Pool Pct. Util

The percentage of pages used to write changes to the recovery log after a transaction is committed. A value below 10% means that the recovery log buffer pool size is adequate. If the percentage increases, consider increasing the recovery log buffer pool size.

Log Pool Pct. Wait

The percentage of requests for pages that are not available because all pages are waiting to write to the recovery log.

If this value is greater than 0, increase the recovery log buffer pool size.

Step 2: Adjust the Recovery Log Buffer Pool

Use the LOGPOOLSIZE server option to set the size of the recovery log buffer pool.

Reorganizing the Database

Over time, database volumes become fragmented. You can restore the efficiency of the database and improve database performance by reorganizing the database using database unload and reload processing. By reloading the database, you compress and reorganize it.

Procedure: Reorganizing the Database

Attention: Before you begin this procedure, perform a backup of your database. If an outage occurs while you are loading and reloading your database, you can use your backup copy for recovering the database.

The DSMSERV UNLOADDB operation assumes that the Tivoli Storage Manager database is usable and reads device information from the database, not from the device configuration file. A database dump operation (DSMSERV DUMPDB), on the other hand, does not assume a usable database and reads from the device configuration file.

To reorganize the database, follow these steps:

1. Ensure that a current device configuration file exists. This file contains a copy of the device class, library, and drive definitions. These definitions are needed for the DSMSERV LOADDB utility. You must specify the name of the device configuration file by using the DEVCONFIG option in the server options file. See "Saving the Device Configuration File" on page 516. Also see *Administrator's Reference* for details on the option and the command.
2. Before unloading the database, estimate how many tapes you will need:
   - If the server is *not* running, use the size of your existing physical database volumes as an estimate of how many tapes to use.
   - If the server is running, you can use the following steps to estimate the number of tapes required:
     a. Request information about the database by using the following command:
        ```
        query db
        ```
     b. Using the output of the QUERY DB command, multiply the *Used Pages* by the *Page Size* to determine space occupied by the database.
     c. Use the result to estimate the number of tapes of a specific device class that you will need to unload the database. The space required will likely be less than your estimate.
3. Halt the server if it is still running.
4. With the server not running, issue the DSMSERV UNLOADDB utility to unload the database to tape. For example:
   ```
   dsmserv unloaddb devclass=tapeclass scratch=yes
   ```

   Note: Keep track of the order in which the tape volumes are written when the database is unloaded. You must specify the volume names in the same order when you reload the database using the DSMSERV LOADDB utility. For this task, you can either:
   - Review the output generated by the DSMSERV UNLOADDB utility and record the order of the volumes.
   - Manually view the volume history file to identify the tape volumes containing the unloaded database. The volumes have a volume type of DBDUMP. See "Saving the Volume History File" on page 514 for details. (Do *not* restart the server and issue QUERY VOLHISTORY at this step.)
5. Format the database and recovery log. For example:
   ```
   dsmserv loadformat 2 logvol1 logvol2 1 dbvol1
   ```

   This utility prepares the existing server database for the DSMSERV LOADDB utility.

6. Reload the database using the volumes that contain the data from the unload operation. For example:

   ```
   dsmserv loaddb devclass=tapeclass volumenames=db001,db002,db003
   ```

   For the volume names, ensure that you do the following:
   - Enter the volume names in the same order in which they were used for the DSMSERV UNLOADDB utility.
   - Separate the volume names with a comma and no intervening spaces.

Chapter 19. Monitoring the Tivoli Storage Manager Server

Administrators can monitor the Tivoli Storage Manager server:

- To find the status of operations
- To display information about objects
- To monitor the record of activity
- To select the types of events to save
- To select a location to save events

See the following sections:

Tasks:
"Using Tivoli Storage Manager Queries to Display Information"
"Using SQL to Query the Tivoli Storage Manager Database" on page 406
"Using the Tivoli Storage Manager Activity Log" on page 411
"Logging Tivoli Storage Manager Events to Receivers" on page 413
"Monitoring Tivoli Storage Manager Accounting Records" on page 427

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Using Tivoli Storage Manager Queries to Display Information

Tivoli Storage Manager provides a variety of QUERY commands that display formatted information about definitions, settings, processes, and status. In some cases, you can display the information in either of two formats: standard or detailed. The standard format presents partial information and is useful in displaying an overview of many objects.

Note: For information about creating customized queries of the database, see "Using SQL to Query the Tivoli Storage Manager Database" on page 406.

Requesting Information about Tivoli Storage Manager Definitions

During TSM system setup, an administrator can define many objects. These objects include storage management policies, database and recovery log volumes, storage pools, and device classes. Tivoli Storage Manager provides queries that display information about these objects.

Most of these definition queries let you request standard format or detailed format. Standard format limits the information and usually displays it as one line per object. Use the standard format when you want to query many objects, for example, all registered client nodes. Detailed format displays the default and

© Copyright IBM Corp. 1993, 2002

specific definition parameters. Use the detailed format when you want to see all the information about a limited number of objects.

Here is an example of the standard output for the QUERY NODES command:

```
Node Name   Platform  Policy       Days      Days  Locked?
                      Domain      Since     Since
                      Name         Last  Password
                                 Access       Set
----------  --------  ---------  ------  --------  -------
CLIENT1     (?)       STANDARD        6         6    No
GEORGE      OS/2      STANDARD        1         1    No
JANET       (?)       STANDARD        1         1    No
JOEOS2      OS/2      STANDARD       <1        <1    No
TOMC        (?)       STANDARD        1         1    No
```

Here is an example of the detailed output for the QUERY NODES command:

```
                     Node Name: JOEOS2
                      Platform: OS/2
            Policy Domain Name: STANDARD
         Last Access Date/Time: 05/19/2000 09:59:52
        Days Since Last Access: 2
        Password Set Date/Time: 05/18/2000 11:04:45
       Days Since Password Set: 3
        Invalid Password Limit:
       Minimum Password Length:
                       Locked?: No
                       Contact:
                   Compression: No
       Archive Delete Allowed?: Yes
        Backup Delete Allowed?: No
        Registration Date/Time: 05/18/2000 11:04:45
     Registering Administrator: OPEN_REGISTRATION
Last Communication Method Used: Tcp/Ip
   Bytes Received Last Session: 226
       Bytes Sent Last Session: 556
Duration of Last Session (sec): 3.32
   Pct. Idle Wait Last Session: 88.48
  Pct. Comm. Wait Last Session: 6.63
  Pct. Media Wait Last Session: 0.00
```

Requesting Information about Client Sessions

When administrators or users access Tivoli Storage Manager, an administrative or client node session is established with the server. The server assigns each client session a unique session number.

To request information about client sessions, enter:

```
query session
```

Figure 59 on page 403 shows a sample client session report.

```
  Sess Comm.  Sess     Wait   Bytes   Bytes Sess  Platform Client Name
Number Method State    Time    Sent   Recvd Type
------ ------ ------ ------ ------- ------- ----- -------- --------------------
     3 Tcp/Ip IdleW    9 S    7.8 K     706 Admin OS/2     TOMC
     5 Tcp/Ip IdleW    0 S    1.2 K     222 Admin OS/2     GUEST
     6 Tcp/Ip Run      0 S      117     130 Admin OS/2     MARIE
```

Figure 59. Information about Client Sessions

Check the *wait time* and *session state*. The *wait time* determines the length of time (seconds, minutes, hours) the server has been in the current state. The *session state* can be one of the following:

Start Connecting with a client session.

Run Running a client request.

End Ending a client session.

RecvW
Waiting to receive an expected message from the client while a database transaction is in progress. A session in this state is subject to the COMMTIMEOUT limit.

SendW
Waiting for acknowledgment that the client has received a message sent by the server.

MediaW
Waiting for removable media to become available.

IdleW Waiting for communication from the client, and a database transaction is NOT in progress. A session in this state is subject to the IDLETIMEOUT limit.

For example, Tivoli Storage Manager cancels the client session if the IDLETIMEOUT option is set to 30 minutes, and a user does not initiate any operations within those 30 minutes. The client session is automatically reconnected to the server when it starts to send data again.

Requesting Information about Server Processes

Most commands run in the foreground, but others generate background processes. In some cases, you can specify that a process run in the foreground. TSM issues messages that provide information about the start and end of processes. In addition, you can request information about active background processes. If you know the process ID number, you can use the number to limit the search. However, if you do not know the process ID, you can display information about all background processes by entering:

```
query process
```

Figure 60 on page 404 shows a server background process report after a DELETE FILESPACE command was issued. The report displays a process ID number, a description, and a completion status for each background process.

```
 Process Process Description      Status
  Number
-------- ------------------------ ---------------------------------------------
       2 DELETE FILESPACE         Deleting filespace DRIVE_D for node CLIENT1:
                                   172 files deleted.
```

Figure 60. Information about Background Processes

Requesting Information about Server Settings

Any administrator can request general server information, most of which is defined by SET commands. To request this information, enter:

```
query status
```

The displayed information includes:

- The server name
- When the server was installed and last started
- Whether the server is enabled or disabled
- Whether client registration is open or closed
- Whether passwords are required for client/server authentication
- How long passwords are valid
- Whether accounting records are being generated
- How long messages remain in the activity log before being deleted
- How many client sessions can concurrently communicate with the server
- How many client node sessions are available for scheduled work
- What percentage of the scheduling start-up window is randomized
- What scheduling mode is being used
- How frequently client nodes can poll for scheduled work
- How many times and how often a client node can retry a failed attempt to perform a scheduled operation
- How long event records remain in the database
- The interval before re-authentication is required for the Web administrative client interface

Querying Server Options

Task	Required Privilege Class
Query server options	Any administrator

Use the QUERY OPTION command to display information about one or more server options.

You can issue the QUERY OPTION command with no operands to display general information about all defined server options. You also can issue the QUERY OPTION command with a specific option name or pattern-matching expression to display information on one or more server options.

To display general information about all defined server options, enter:

```
query option
```

You can set options by editing the server options file. See *Administrator's Reference* for more information.

Querying the System

The QUERY SYSTEM command lets you combine multiple queries of your Tivoli Storage Manager system into a single command. This command can be used to collect statistics and to provide information for problem analysis by IBM service. When you issue the QUERY SYSTEM command, the server issues the following queries:

QUERY ASSOCIATION
Displays all client nodes that are associated with one or more client schedules

QUERY COPYGROUP
Displays all backup and archive copy groups (standard format)

QUERY DB
Displays information about the database (detailed format)

QUERY DBVOLUME
Displays information about all database volumes (detailed format)

QUERY DEVCLASS
Displays all device classes (detailed format)

QUERY DOMAIN
Displays all policy domains (standard format)

QUERY LOG
Displays information about the recovery log (detailed format)

QUERY LOGVOLUME
Displays information about all recovery log volumes (detailed format)

QUERY MGMTCLASS
Displays all management classes (standard format)

QUERY OPTION
Displays all server options

QUERY PROCESS
Displays information about all active background processes

QUERY SCHEDULE
Displays client schedules (standard format)

QUERY SESSION
Displays information about all administrative and client node sessions in standard format

QUERY STATUS
Displays general server parameters, such as those defined by SET commands

QUERY STGPOOL
Displays information about all storage pools (detailed format)

QUERY VOLUME
Displays information about all storage pool volumes (standard format)

SELECT
Displays the results of two SQL queries:

```
select platform_name,count(*) from nodes group by platform_name
select stgpool_name,devclass_name,count(*) from volumes
group by stgpool_name,devclass_name
```

The first command displays the number of client nodes by platform.

The second command displays the name and associated device class of all storage pools having one or more volumes assigned to them.

Using SQL to Query the Tivoli Storage Manager Database

You can use a standard SQL SELECT statement to get information from the database. The SELECT command is a subset of the SQL92 and SQL93 standards.

Tivoli Storage Manager also provides an open database connectivity (ODBC) driver. The driver allows you to use a relational database product such as Lotus Approach® to query the database and display the results.

Using the ODBC Driver

Tivoli Storage Manager provides an ODBC driver for Windows. The driver supports the ODBC Version 2.5 application programming interface (API). Because TSM supports only the SQL SELECT statement (query), the driver does not conform to any ODBC API or SQL grammar conformance level. After you install this driver, you can use a spreadsheet or database application that complies with ODBC to access the database for information.

The ODBC driver set-up is included in the client installation package. The client installation program can install the ODBC driver and set the corresponding registry values for the driver and data sources. For more information on setting up the ODBC driver, see *Installing the Clients*.

To open the database through an ODBC application, you must log on to the server (the defined data source). Use the name and password of a registered administrator. After you log on to the server, you can perform query functions provided by the ODBC application to access database information.

Issuing SELECT Commands

You can issue the SELECT command from the command line of an administrative client. You cannot issue this command from the server console.

The SELECT command supports a subset of the syntax of the SELECT statement as documented in the SQL92 and SQL93 standards. For complete information about how to use the SELECT statement, refer to these standards or to other publications about SQL.

Issuing the SELECT command to the server can use a significant amount of server resources to run the query. Complicated queries or queries that run for a long time can interfere with normal server operations. If your query requires excessive server resource to generate the results, you will receive a message asking you to confirm that you wish to continue.

Note: To allow any use of the SELECT command, the database must have at least 4MB of free space. For complex queries that require significant processing, additional free space is required in the database. See "Exhausting Temporary Table Storage" on page 408 for details.

Learning What Information Is Available: System Catalog Tables

To help you find what information is available in the database, Tivoli Storage Manager provides three system catalog tables:

SYSCAT.TABLES
Contains information about all tables that can be queried with the SELECT command.

SYSCAT.COLUMNS
Describes the columns in each table.

SYSCAT.ENUMTYPES
Defines the valid values for each enumerated type and the order of the values for each type.

You can issue the SELECT command to query these tables to determine the location of the information that you want. For example, to get a list of all tables available for querying in the database, enter the following command:

```
select * from syscat.tables
```

The following shows part of the results of this command:

```
     TABSCHEMA: TSM
       TABNAME: ACTLOG
   CREATE_TIME:
      COLCOUNT: 11
INDEX_COLCOUNT: 1
  UNIQUE_INDEX: FALSE
       REMARKS: Server activity log

     TABSCHEMA: TSM
       TABNAME: ADMINS
   CREATE_TIME:
      COLCOUNT: 17
INDEX_COLCOUNT: 1
  UNIQUE_INDEX: TRUE
       REMARKS: Server administrators

     TABSCHEMA: TSM
       TABNAME: ADMIN_SCHEDULES
   CREATE_TIME:
      COLCOUNT: 15
INDEX_COLCOUNT: 1
  UNIQUE_INDEX: TRUE
       REMARKS: Administrative command schedules

     TABSCHEMA: TSM
       TABNAME: ARCHIVES
   CREATE_TIME:
      COLCOUNT: 10
INDEX_COLCOUNT: 5
  UNIQUE_INDEX: FALSE
       REMARKS: Client archive files
```

Examples

The SELECT command lets you customize a wide variety of queries. This section shows two examples. For many more examples of the command, see the *Administrator's Reference*.

Example 1: Find the number of nodes by type of operating system by issuing the following command:

```
select platform_name,count(*) as "Number of Nodes" from nodes
group by platform_name
```

This command gives results like the following:

```
PLATFORM_NAME     Number of Nodes
-------------     ---------------
OS/2                           45
AIX                            90
Windows                        35
```

Example 2: For all active client sessions, determine how long they have been connected and their effective throughput in bytes per second:

```
select session_id as "Session", client_name as "Client", state as "State",
  current_timestamp-start_time as "Elapsed Time",
  (cast(bytes_sent as decimal(18,0)) /
  cast((current_timestamp-start_time)seconds as decimal(18,0)))
  as "Bytes sent/second",
  (cast(bytes_received as decimal(18,0)) /
  cast((current_timestamp-start_time)seconds as decimal(18,0)))
  as "Bytes received/second"
  from sessions
```

This command gives results like the following:

```
                Session: 24
                 Client: ALBERT
                  State: Run
           Elapsed Time: 0 01:14:05.000000
      Bytes sent/second: 564321.9302768451
  Bytes received/second: 0.0026748857944

                Session: 26
                 Client: MILTON
                  State: Run
           Elapsed Time: 0 00:06:13.000000
      Bytes sent/second: 1638.5284210992221
  Bytes received/second: 675821.6888561849
```

Exhausting Temporary Table Storage

SQL SELECT queries run from temporary table storage in the database. At least a 4MB partition must be available in the database for this purpose. Without this partition, temporary table storage space will become exhausted, and the SELECT query will no longer run.

To determine how much temporary table storage space is available in your database, issue the QUERY DB command. The server displays a report, like the following:

```
Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size     Pages     Pages       %Util
     (MB)     (MB)      (MB)      (MB) (bytes)
--------- -------- --------- --------- ------- --------- --------- ----- -----
        8        4         4         0   4,096     1,024        94   9.3   9.2
```

Check the value in the **Maximum Reduction** field. If this field shows a value of at least 4MB, you can perform SELECT queries.

If the **Maximum Reduction** value is below 4MB, you will not be able to perform SELECT queries. The database is either full or fragmented.

- If the database is full, increase the size of the database. See “Increasing the Size of the Database or Recovery Log” on page 388 for details.
- If the database is fragmented, either add a volume or unload and load your database. See “Reorganizing the Database” on page 397 for details.

Note: Complex SELECT queries (for example, those including the ORDER BY clause, the GROUP BY clause, or the DISTINCT operator) may require more than 4MB temporary table storage space.

Using SELECT Commands in Tivoli Storage Manager Scripts

A Tivoli Storage Manager script is one or more commands that are stored as an object in the database. You can run a script from an administrative client, the web interface, or the server console. You can also include it in an administrative command schedule to run automatically. See “Tivoli Storage Manager Server Scripts” on page 372 for details. You can define a script that contains one or more SELECT commands. Tivoli Storage Manager is shipped with a file that contains a number of sample scripts. The file, *scripts.smp*, is in the server directory. To create and store the scripts as objects in your server’s database, issue the DSMSERV RUNFILE command during installation:

```
> dsmserv runfile scripts.smp
```

You can also run the file as a macro from an administrative command line client:

```
macro scripts.smp
```

The sample scripts file contains TSM commands. These commands first delete any scripts with the same names as those to be defined, then define the scripts. The majority of the samples create SELECT commands, but others do such things as define and extend database volumes and back up storage pools. You can also copy and change the sample scripts file to create your own scripts.

Here are a few examples from the sample scripts file:

```
def script q_inactive_days '/* ------------------------------------------*/'
upd script q_inactive_days '/* Script Name:  Q_INACTIVE                  */'
upd script q_inactive_days '/* Description: Display nodes that have not  */'
upd script q_inactive_days '/*              accessed TSM for a           */'
upd script q_inactive_days '/*              specified number of days     */'
upd script q_inactive_days '/* Parameter 1: days                         */'
upd script q_inactive_days '/* Example:     run q_inactive_days 5        */'
upd script q_inactive_days '/* ------------------------------------------*/'
upd script q_inactive_days "select node_name,lastacc_time from nodes where -"
upd script q_inactive_days " cast((current_timestamp-lastacc_time)days as -"
upd script q_inactive_days " decimal) >= $1 "

/* Define a DB volume and extend the database                           */

def script def_db_extend '/*  -----------------------------------------*/'
upd script def_db_extend '/*  Script Name:  DEF_DB_EXTEND              */'
upd script def_db_extend '/*  Description: Define a database volume,   */'
upd script def_db_extend '/*               and extend the database     */'
upd script def_db_extend '/*  Parameter 1: db volume name              */'
upd script def_db_extend '/*  Parameter 2: extension megabytes         */'
upd script def_db_extend '/*  Example:  run def_db_extend VOLNAME 12   */'
upd script def_db_extend '/*  -----------------------------------------*/'
upd script def_db_extend ' def dbv  $1 '
upd script def_db_extend ' if (rc_ok) extend db $2'
upd script def_db_extend ' if (warning, error) q db f=d'
```

Canceling a SELECT Command

If a SELECT command will require a significant amount of resources, the server asks if you want to continue. You can cancel the command at that time. Cancel the command from the console session or an administrative client session.

Controlling the Format of SELECT Results

Tivoli Storage Manager provides commands to control the format of results of SELECT commands. You can control:

- How SQL data types such as VARCHAR are displayed, in wide or narrow format (SET SQLDISPLAYMODE)
- The format of date and time values in the results (SET SQLDATETIMEFORMAT)
- Whether SQL arithmetic results are truncated or rounded (SET SQLMATHMODE)

Note: Using the SET commands to change these settings keeps the settings in effect only for the current administrative client session. You can query these settings by using the QUERY SQLSESSION command.

Querying the SQL Activity Summary Table

You can query the SQL activity summary table to view statistics about each client session and server process. For a listing of the column names and their descriptions from the activity summary table, enter the following command:

```
select colname,remarks from columns where tabname='summary'
```

Here are a few example queries of the activity summary table.

- To display all events starting at 00:00 a.m. of the current day until the present time, enter:

  ```
  select * from summary
  ```

  The result might look like this:

  ```
  START_TIME: 2000-07-22 19:32:00.000000
    END_TIME: 2000-07-22 19:32:56.000000
    ACTIVITY: BACKUP
      NUMBER: 43
      ENTITY: DWE
    COMMMETH: Named Pi
     ADDRESS:
    EXAMINED: 7
    AFFECTED: 7
      FAILED: 0
       BYTES: 2882311
        IDLE: 51
      MEDIAW: 0
   PROCESSES: 1
  SUCCESSFUL: YES

  ANS8002I Highest return code was 0.
  ```

- To display all events starting at or after 00:00 a.m. on September 24, 2000 until the present time, enter:

  ```
  select * from summary where start_time>= '2000-09-24 00:00'
  ```

You can determine how long to keep information in the summary table. For example, to keep the information for 5 days, enter the following command:

```
set summaryretention 5
```

To keep no information in the table, specify a value of 0.

Creating Output for Use by Another Application

You can redirect the output of SELECT commands to a file in the same way as you would redirect the output of any command. When redirecting this output for use in another program (for example, a spreadsheet or database program), write the output in a format easily processed by the program to be used.

Two standard formats for tabular data files are *comma-separated values* (CSV) and *tab-separated values* (TSV). Most modern applications that can import tabular data can read one or both of these formats.

Use the administrative client command line options -COMMADELIMITED or -TABDELIMITED to select one of these formats for tabular query output. All tabular output during the administrative session will be formatted into either comma-separated or tab-separated values. For details about using command line options, see the *Administrator's Reference*.

The use of command output redirection and one of the delimited output format options lets you create queries whose output can be further processed in other applications. For example, based on the output of a SELECT command, a spreadsheet program could produce graphs of average file sizes and file counts summarized by type of client platform.

For details about redirecting command output, see the *Administrator's Reference*.

Using the Tivoli Storage Manager Activity Log

Task	**Required Privilege Class**
Request information from the activity log	Any administrator
Set the activity log retention period	System
Change the size of the activity log	System or unrestricted storage

The activity log contains all messages normally sent to the server console during server operation. The only exceptions are responses to commands entered at the console, such as responses to QUERY commands.

Examples of messages sent to the activity log include:
- When client sessions start or end
- When migration starts and ends
- When backup versions expire
- What data is exported to tape
- When expiration processing is performed
- What export or import processing is performed

Any error messages sent to the server console are also stored in the activity log.

Use the following sections to adjust the size of the activity log, set an activity log retention period, and request information about the activity log.

Requesting Information from the Activity Log

You can request information stored in the activity log. To minimize processing time when querying the activity log, you can:

- Specify a time period in which messages have been generated. The default for the QUERY ACTLOG command shows all activities that have occurred in the previous hour.
- Specify the message number of a specific message or set of messages.
- Specify a string expression to search for specific text in messages.
- Specify the QUERY ACTLOG command from the command line for large queries instead of using the graphical user interface.
- Specify whether the originator is the server or client. If it is the client, you can specify the node, owner, schedule, domain, or session number. If you are doing client event logging to the activity log and are only interested in server events, then specifying the server as the originator will greatly reduce the size of the results.

For example, to review messages generated on May 30 between 8 a.m. and 5 p.m., enter:

```
query actlog begindate=05/30/2000 enddate=05/30/2000
begintime=08:00 endtime=17:00
```

To request information about messages related to the expiration of files from the server storage inventory, enter:

```
query actlog msgno=0813
```

Refer to *Messages* for message numbers.

You can also request information only about messages logged by one or all clients. For example, to search the activity log for messages from the client for node JEE:

```
query actlog originator=client node=jee
```

Setting the Activity Log Retention Period

Use the SET ACTLOGRETENTION command to specify how long activity log information is kept in the database. The server automatically deletes messages from the activity log once the day that was specified with the SET ACTLOGRETENTION command has passed. At installation, the activity log retention period is set to one day. To change the retention period to 10 days, for example, enter:

```
set actlogretention 10
```

To disable activity log retention, set the SET ACTLOGRETENTION command to zero. To display the current retention period for the activity log, query the server status.

Changing the Size of the Activity Log

Because the activity log is stored in the database, the size of the activity log should be factored into the amount of space allocated for the database. Allow at least 1MB of additional space for the activity log.

The size of your activity log depends on how many messages are generated by daily processing operations and how long you want to retain those messages in the activity log. When retention time is increased, the amount of accumulated data also increases, requiring additional database storage.

When there is not enough space in the database or recovery log for activity log records, the server stops recording and sends messages to the server console. If you increase the size of the database or recovery log, the server starts activity log recording again.

If you do not have enough space in the database for the activity log, you can do one of the following:

- Allocate more space to the database
- Reduce the length of time that messages are kept in the activity log

For information about increasing the size of the database or recovery log, see "Increasing the Size of the Database or Recovery Log" on page 388.

Logging Tivoli Storage Manager Events to Receivers

The server and client messages provide a record of TSM activity that you can use to monitor the server. You can log server messages and most client messages as *events* to one or more repositories called *receivers*. You can log the events to any combination of the following receivers:

TSM server console and activity log
See "Logging Events to the Tivoli Storage Manager Server Console and Activity Log" on page 415.

File and user exits
See "Logging Events to a File Exit and a User Exit" on page 415.

Tivoli event console
See "Logging Events to the Tivoli/Enterprise Console" on page 417.

Simple Network Management Protocol (SNMP)
See "Logging Events to an SNMP Manager" on page 418.

Event server receiver (Enterprise Event Logging)
Routes the events to an event server. See "Enterprise Event Logging: Logging Events to Another Server" on page 423.

In addition, you can filter the types of events to be enabled for logging. For example, you might enable only severe messages to the event server receiver and one or more specific messages, by number, to another receiver. Figure 61 on page 414 shows a possible configuration in which both server and client messages are filtered by the event rules and logged to a set of specified receivers.

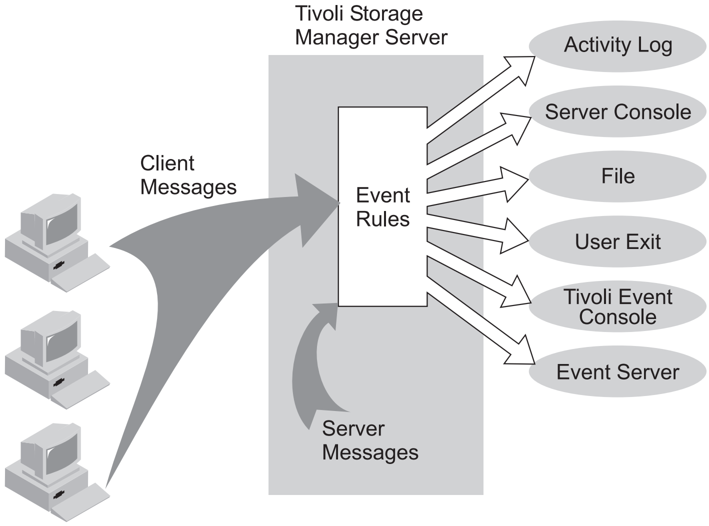


Figure 61. Event Logging Overview

Task	Required Privilege Class
Enable or disable events Begin or end event logging	System

Controlling Event Logging

To control event logging do the following:

1. Enable or disable logging for one or more event types and for one or more receivers.
2. Begin or end logging to one or more receivers.

Enabling and Disabling Events

When you enable or disable events, you can specify the following:

- A message number or an event severity (ALL, INFO, WARNING, ERROR, or SEVERE).
- Events for one or more client nodes (NODENAME) or for one or more servers (SERVERNAME).

To enable or disable events, issue the ENABLE EVENTS and DISABLE EVENTS commands. For example,

- To enable event logging to a user exit for all error and severe server messages, enter:

  ```
  enable events userexit error,severe
  ```

- To enable event logging to a user exit for severe client messages for all client nodes, enter:

  ```
  enable events userexit severe nodename=*
  ```

- To disable event logging to a user exit for error server messages, enter

  ```
  disable events userexit error
  ```

If you specify a receiver that is not supported on any platform, or if you specify an invalid event or name, TSM issues an error message. However, any valid receivers, events, or names that you specified are still enabled. Certain events, such as messages that are issued during server start-up and shutdown, automatically go to the console. They do not go to other receivers, even if they are enabled.

Note: Server messages in the SEVERE category and message ANR9999 can provide valuable diagnostic information if there is a serious problem. For this reason, you should not disable these messages. Use the SET CONTEXTMESSAGING ON command to get additional information that could help determine the cause of ANR9999D messages. The Tivoli Storage Manager polls the server components for information that includes process name, thread name, session ID, transaction data, locks that are held, and database tables that are in use.

Beginning and Ending Event Logging

A receiver for which event logging has begun is an *active receiver*. To begin and end logging for one or more receivers, issue the BEGIN EVENTLOGGING and END EVENTLOGGING commands.

At server start-up event logging begins automatically to the server console and activity log and for any receivers that are started based on entries in the server options file. See the appropriate receiver sections for details. To begin logging events to receivers for which event logging is not started automatically, issue the BEGIN EVENTLOGGING command. You can also use this command after you have disabled event logging to one or more receivers. To end event logging for an active receiver issue the END EVENTLOGGING command.

For example,

- To begin logging events to the event server, enter:
  ```
  begin eventlogging eventserver
  ```
- To end logging events to the event server, enter:
  ```
  end eventlogging eventserver
  ```

Logging Events to the Tivoli Storage Manager Server Console and Activity Log

Logging events to the server console and activity log begins automatically at server startup. To enable all error and severe client events to the console and activity log, issue the following command:

```
enable events console,actlog error,severe
```

Note: Enabling client events to the activity log will increase the database utilization. You can set a retention period for the log records by using the SET ACTLOGRETENTION command (see "Setting the Activity Log Retention Period" on page 412). At server installation, this value is set to one day. If you increase the retention period, utilization is further increased. For more information about the activity log, see "Using the Tivoli Storage Manager Activity Log" on page 411.

You can disable server and client events to the server console and client events to the activity log. However, you cannot disable server events to the activity log. Also, certain messages, such as those issued during server startup and shutdown and responses to administrative commands, will still be displayed at the console even if disabled.

Logging Events to a File Exit and a User Exit

You can log events to a file exit and a user exit:

- A file exit is a file that receives all the information related to its enabled events. Be aware that this file can rapidly grow in size depending on the events enabled for it. There are two versions of the file exit: binary and text. The binary file exit stores each logged event as a record, while the text file exit stores each logged event as a fixed-sized, readable line. For more information about the text file exit, see "Readable Text File Exit (FILETEXTEXIT) Format" on page 618.

- A user exit is an external interface in the form of an executable, user-written program. TSM supports user exits.

Note: Both types of event receivers must be specified in the server options file (dsmserv.opt) file.

Both file and user exits receive event data in the same data block structure. Setting up logging for these receivers is also similar:

1. Add an option for the exit to the server options file:
   - **For a file exit:** Add either the FILEEXIT option (for a binary file exit) or FILETEXTEXIT (for a text file exit) option.
     - Specify whether event logging to the file exit receiver begins automatically at server startup. The parameters are YES and NO. If you do not specify YES, you must begin event logging manually by issuing the BEGIN EVENTLOGGING command.
     - Specify the file where each logged event is to be stored.
     - Specify how files will be stored if the file being stored already exists. REPLACE will overwrite the existing file, APPEND will append data to the existing file, and PRESERVE will not overwrite the existing file.

     For example,

     ```
     fileexit yes /tsm/server/data replace
     ```

     ```
     filetextexit yes /tsm/server/data replace
     ```
   - **For a user exit:** Add the USEREXIT option.
     - Specify whether event logging to the user exit receiver begins automatically at server startup. The parameters for this option are YES and NO. If you do not specify YES, you must begin event logging manually by issuing the BEGIN EVENTLOGGING command.
     - Specify a module name of the user exit. This is the name of a shared library containing the exit.

     For example,

     ```
     userexit no fevent.exit
     ```
2. Enable events for the receiver. You must specify the name of the user exit in the USEREXIT server option and the name of the file in the FILEEXIT server option. Here are two examples:

   ```
   enable events file error
   ```

   ```
   enable events userexit error,severe
   ```

   You can also enable events to one or more client nodes or servers by specify the NODENAME OR SERVERNAME parameter. See "Enabling and Disabling Events" on page 414 for more information.
3. If you did not specify YES in the server option, begin event logging. For example, to begin event logging for a user-defined exit, enter:

   ```
   begin eventlogging userexit
   ```

   See "Beginning and Ending Event Logging" on page 415 for more information.

Logging Events to the Tivoli/Enterprise Console

TSM includes the Tivoli receiver, a Tivoli/Enterprise Console (T/EC) adapter for sending events to the T/EC. You can specify the events to be logged based on their source. The valid event names are:

Event Name	Source
TSM_SERVER_EVENT	TSM server
TSM_CLIENT_EVENT	TSM clients
TSM_APPL_EVENT	TSM application program interface
TSM_TDP_DOMINO_EVENT	TDP for Domino
TSM_TDP_EXCHANGE_EVENT	TDP for MS Exchange
TSM_TDP_INFORMIX_EVENT	TDP for Informix
TSM_TDP_ORACLE_EVENT	TDP for Oracle
TSM_TDP_SQL_EVENT	TDP for MS SQL

The application client must have enhanced T/EC support enabled in order to route the events to the T/EC. Because of the number of messages, you should not enable all messages from a node to be logged to the T/EC.

To set up Tivoli as a receiver for event logging:

1. Define the TSM event classes to the T/EC with the *ibmtsm.baroc* file, which is distributed with the server.

   Note: Please refer to TEC documentation for instruction on removing an existing baroc file, if needed, and installing a new baroc file. If you have migrated from ADSM Version 3 and have an existing *ibmadsm.baroc* file, do one of the following:

   - Remove the file.
   - Create a new rule base.
   - Copy the file.

   Before the events are displayed on a T/EC, you must import *ibmtsm.baroc* into an existing rule base or create a new rule base and activate it. To do this:

   - From the TME® desktop, click on the **Rule Base** icon to display the pop-up menu.
   - Select **Import**, then specify the location of the *ibmtsm.baroc* file.
   - Select the **Compile** pop-up menu.
   - Select the **Load** pop-up menu and **Load, but activate only when server restarts** from the resulting dialog.
   - Shut down the event server and restart it.

   To create a new rule base, do the following:

   a. Click on the **Event Server** icon from the TME desktop. The **Event Server Rules Bases** window will open.
   b. Select **Rule Base** from the **Create** menu.
   c. Optionally, copy the contents of an existing rule base into the new rule base by selecting the **Copy** pop-up menu from the rule base to be copied.
   d. Click on the **RuleBase** icon to display the pop-up menu.
   e. Select **Import** and specify the location of the *ibmtsm.baroc* file.
   f. Select the **Compile** pop-up menu.

   g. Select the **Load** pop-up menu and **Load, but activate only when server restarts** from the resulting dialog.
   h. Shut down the event server and restart it.
2. To define an event source and an event group:
   a. From the TME desktop, select **Source** from the **EventServer** pop-up menu. Define a new source whose name is TSM from the resulting dialog.
   b. From the TME desktop, select **Event Groups** from the **EventServer** pop-up menu. From the resulting dialog, define a new event group for TSM and a filter that includes event classes IBMTSMSERVER_EVENT and IBMTSMCLIENT_EVENT.
   c. Select the **Assign Event Group** pop-up menu item from the **Event Console** icon and assign the new event group to the event console.
   d. Double-click on the **Event Console** icon to start the configured event console.
3. Enable events for logging to the Tivoli receiver. See "Enabling and Disabling Events" on page 414 for more information.
4. In the server options file *(dsmserv.opt)*, specify the location of the host on which the Tivoli server is running. For example, to specify a Tivoli server at the IP address 9.114.22.345:1555, enter the following:

   ```
   techost 9.114.22.345
   tecport 1555
   ```
5. Begin event logging for the Tivoli receiver. You do this in one of two ways:
   - To begin event logging automatically at server start up, specify the following server option:

     ```
     tecbegineventlogging yes
     ```

   Or
   - Enter the following command:

     ```
     begin eventlogging tivoli
     ```

     See "Beginning and Ending Event Logging" on page 415 for more information.

Logging Events to an SNMP Manager

You can use the simple network management protocol (SNMP) together with event logging to do the following:

- Set up an SNMP heartbeat monitor to regularly check that the TSM server is running.
- Send traps to an SNMP manager, such as NetView® or Tivoli.
- Run TSM scripts and retrieve output and return codes. See "Tivoli Storage Manager Server Scripts" on page 372 for details.

The management information base (MIB), which is shipped with TSM, defines the variables that will run server scripts and return the server scripts' results. You must register SNMPADMIN, the administrative client the server runs these scripts under. Although a password is not required for the subagent to communicate with the server and run scripts, a password should be defined for SNMPADMIN to prevent access to the server from unauthorized users. An SNMP password (community name) is required, however, to access the SNMP agent, which forwards the request to the subagent.

Note: Because the SNMP environment has weak security, you should consider not granting SNMPADMIN any administrative authority. This restricts SNMPADMIN to issuing only TSM queries.

SNMP SET requests are accepted for the name and input variables associated with the script names stored in the MIB by the SNMP subagent. This allows a script to be run by running a GET request for the ibmAdsm1ReturnValue and ibmAdsm2ReturnValue variables. A GETNEXT request will not cause the script to run. Instead, the results of the previous script processed will be retrieved. When an entire table row is retrieved, the GETNEXT request is used. When an individual variable is retrieved, the GET request is used.

Here is a sample TSM configuration with SNMP:

1. A TSM server on System A communicates with a subagent on system B.
2. The subagent on System B communicates with an agent on system C, which forward traps to a Netview system.
3. System D runs a TSM server that also uses the subagent on system B.
4. System B also has a TSM server that uses the subagent on system B.

To run an arbitrary command from an SNMP management application, for example, NetView, follow these steps:

1. Choose the name and parameters for a TSM script.
2. Use the application to communicate with the SNMP agent. This agent changes the TSM MIB variable for one of the two script names that the TSM subagent maintains. The SNMP agent also sets the parameter variables for one of the two scripts.
3. Use the application to retrieve the variable *ibmAdsmReturnValue1.x* or *ibmAdsmReturnValue2.x*, where *x* is the index of the server that is registered with the subagent.

To set the variables associated with the script (for example, *ibmAdsmServerScript1/2* or *ibmAdsmM1Parm1/2/3*), the nodes on which the subagent and the agent are run must have read-write authority to the MIB variables. This is done through the SNMP configuration process on the system that the SNMP agent runs on. In AIX, the file name is */etc/snmpd.conf*.

Here is an AIX example:

```
community public 9.115.20.174 255.255.255.254 readWrite
community public 9.115.46.25  255.255.255.254 readWrite
community public 127.0.0.1    255.255.255.254 readWrite
community  public 9.115.20.176 255.255.255.254 readWrite
smux       1.3.6.1.4.1.2.3.1.2.2.1.1.2 public
```

The statements grant read-write authority to the MIB for the local node through the loopback mechanism (127.0.0.1), and to nodes with the three 9.115.xx.xx addresses. On AIX, TSM installation automatically updates the*/etc/mib.defs* file with the names of the TSM MIB variables. This makes the MIB variables available to applications like the AIX System Monitor product. The smux statement allows the dpid2 daemon to communicate with snmpd.

The *snmpinfo* command is shipped with the System Monitor product. Here is an example of this command used to set and retrieve MIB variables:

```
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmServerScript1.1=QuerySessions
```

This command issues the set operation (-ms), passing in community name **public**, sending the command to host **tpcnov73**, and setting up variable *ibmAdsmServerScript1* to have the value *QuerySessions*. *QuerySessions* is the name of a server script that has been defined on a server that will register with the TSM subagent. In this case, the first server that registers with the subagent is the *.1* suffix in *ibmAdsmServerScript1.1*. The following commands set the parameters for use with this script:

```
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmM1Parm1.1=xyz
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmM1Parm2.1=uvw
snmpinfo -v -ms -c public -h tpcnov73 ibmAdsmM1Parm3.1=xxx
```

You can set zero to three parameters. Only the script name is needed. To make the *QuerySessions* script run, retrieve the *ibmAdsmM1ReturnValue* variable (in this case, *ibmAdsmM1ReturnValue.1*. For example:

```
snmpinfo -v -mg -c public -h tpcnov73 ibmAdsmM1ReturnValue.1
```

The results of the command are returned as a single string with embedded carriage return/newline characters.

Note: Not all MIB browsers properly handle embedded carriage return/newline characters.

In this case, *ibmAdsmM1ReturnCode.1* will contain the return code associated with the running of the script. If *ibmAdsmM2ReturnValue* is retrieved, the results of running the script named in *ibmAdsmServerScript2* are returned as a single numeric return code. Notice the *-mg* instead of *-ms* to signify the GET operation in the command to retrieve *ibmAdsmM1ReturnValue.1*. If the entire row is retrieved, the command is not run. Instead, the results from the last time the script was run are retrieved. This would be the case if the following command were issued:

```
snmpinfo -v -md -c public -h tpcnov73 ibmAdsm
```

in which all TSM MIB variables are displayed.

An SNMP agent is needed for communication between an SNMP manager and its managed systems. The SNMP agent is accomplished through the **snmpd daemon**. The Distributed Protocol Interface (DPI®) Version 2 is an extension of this SNMP agent.

TSM management through SNMP requires additional information in the MIB of the local agent. Therefore, an SNMP agent supporting DPI Version 2 must be used to communicate with the TSM subagent. This SNMP agent is not included with TSM. AIX 4.2.1 and later include such an SNMP agent. IBM makes the SystemView® agent available for Windows and AIX. The TSM subagent is included with TSM and, before server startup, must be started as a separate process communicating with the SNMP agent.

The SNMP manager system can reside on the same system as the TSM server, but typically would be on another system connected through SNMP. The SNMP management tool can be any application, such as NetView or Tivoli, that can manage information through SNMP MIB monitoring and traps. The TSM server system runs the processes needed to send TSM event information to an SNMP management system. The processes are:

- SNMP agent (snmpd)
- TSM SNMP subagent (dsmsnmp)
- TSM server (dsmserv)

Cross-system support for communication between the server and subagent is not supported, and these products must be installed and run on the TSM server system. Figure 62 illustrates a typical TSM implementation:

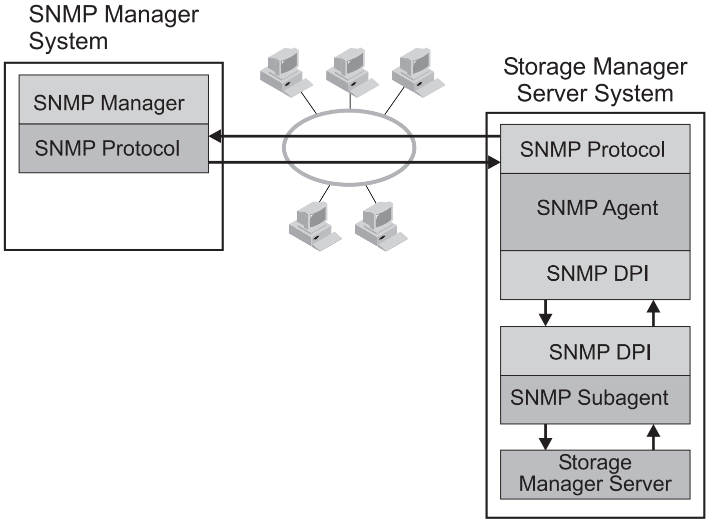


Figure 62. TSM SNMP Implementation

Figure 63 on page 422 shows how the communication for SNMP works in a TSM system:

- The SNMP manager and agent communicate with each other through the SNMP protocol. The SNMP manager passes all requests for variables to the agent.
- The agent then passes the request to the subagent and sends the answer back to the manager. The agent responds to the manager's requests and informs the manager about events by sending traps.
- The agent communicates with both the manager and subagent. It sends queries to the subagent and receives traps that inform the SNMP manager about events taking place on the application monitored through the subagent. The SNMP agent and subagent communicate through the Distributed Protocol Interface (DPI). Communication takes place over a stream connection, which typically is a TCP connection but could be another stream-connected transport mechanism.
- The subagent answers MIB queries of the agent and informs the agent about events by sending traps. The subagent can also create and delete objects or subtrees in the agent's MIB. This allows the subagent to define to the agent all the information needed to monitor the managed application.

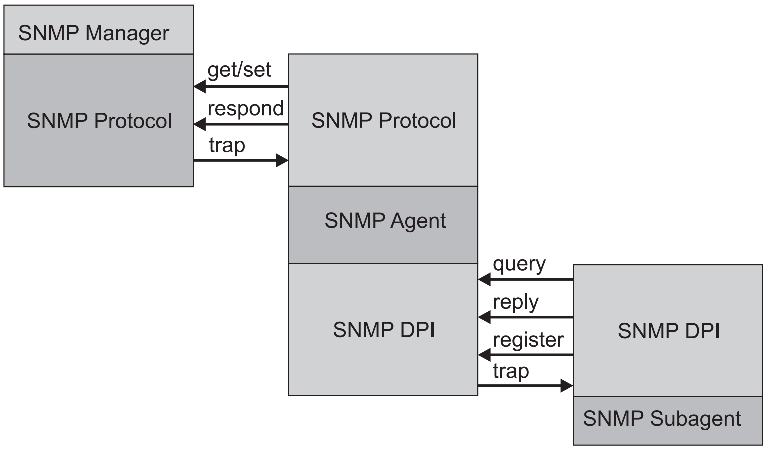


Figure 63. Manager-Agent-Subagent Communication

Notes:

1. You can start *dsmsnmp* and the server in any order. However, starting *dsmsnmp* first is more efficient in that it avoids retries.
2. The MIB file name is *adsmserv.mib*.
3. The AIX install updates */etc/mib.defs*
4. *mib2adsm.tbl* is for concatenating to *mib2.tbl* for Windows SystemView agents.

Configuring Tivoli Storage Manager SNMP

The Tivoli Storage Manager SNMP set up procedure is illustrated by Figure 64:

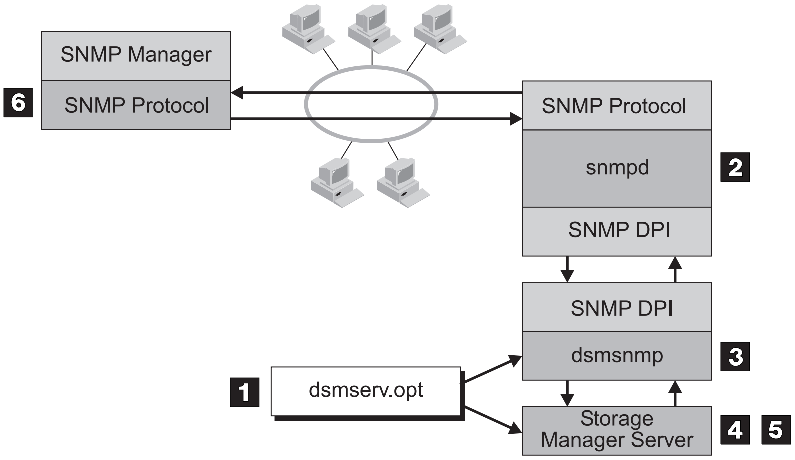


Figure 64. Tivoli Storage Manager SNMP Set Up

To set up TSM monitoring through SNMP, do the following:

1. Modify the server options file to specify the SNMP communication method. Figure 65 on page 423 displays an example of a SNMP communication method setting in the server options file. You must specify the COMMMETHOD option. For details about server options, see the server options section in *Administrator's Reference*.

```
commmethod                snmp
   snmpheartbeatinterval    5
   snmpmessagecategory      severity
```

Figure 65. Example of SNMP Communication Method Options

2. Install, configure, and start the SNMP agent as described in the documentation for that agent. The SNMP agent must support the DPI Version 2.0 standard.

   For example, the AIX SystemView agent is configured by customizing the file */etc/snmpd.conf*. A default configuration might look like this:

   ```
   logging    file=/var/snmp/snmpd.log  enabled
   logging    size=0  level=0
   community  public
   community  private 127.0.0.1   255.255.255.255 readWrite
   community  system  127.0.0.1   255.255.255.255 readWrite  1.17.2
   view       1.17.2 system enterprises view
   trap       public  <snmp_manager_ip_adr>   1.2.3 fe
   snmpd      maxpacket=16000 smuxtimeout=60
   smux       1.3.6.1.4.1.2.3.1.2.2.1.1.2 public
   ```

   where <*snmp_manager_ip_adr*> is the IP address of the system running the SNMP management application.

   Before starting the agent, ensure that the DPI agent has been started and not the default SNMP agent that ships with the operating system or with TCP/IP.

   Note: For AIX 4.2.1 and above, the correct agent is shipped with the system. If you are using the SystemView agent (rather than the version that ships with AIX 4.2.1 and later), you must set the SVA_SNMPD environment variable to ensure that the correct agent is started. You can set the variable to any value. For example, on AIX (korn shell) use the following export command:

   ```
   # export SVA_SNMPD="active"
   ```

   Then run svastart.
3. Start TSM SNMP subagent through the dsmsnmp executable.
4. Start the TSM server to begin communication through the configured TCP/IP port with the subagent.
5. Begin event logging for the SNMP receiver, and enable events to be reported to SNMP. For example, issue the following commands:

   ```
   begin eventlogging snmp
   enable event snmp all
   ```
6. Define the TSM SNMP MIB values for the SNMP manager to help format and display the TSM SNMP MIB variables and messages. The *adsmserv.mib* file ships with the TSM server and must be loaded by the SNMP manager. For example, when you run NetView for OS/2® as an SNMP manager, the *adsmserv.mib* file is copied to the *netview_path**SNMP_MIB* directory and then loaded through the following command:

   ```
   [C:\] loadmib -load adsmserv.mib
   ```

Enterprise Event Logging: Logging Events to Another Server

One or more servers can send server events and events from their own clients to another server for logging. The sending server receives the enabled events and routes them to a designated event server. This is done by a receiver that Tivoli Storage Manager provides. At the event server, an administrator can enable one or more receivers for the events being routed from other servers. Figure 66 shows the relationship of a sending TSM server and a TSM event server.

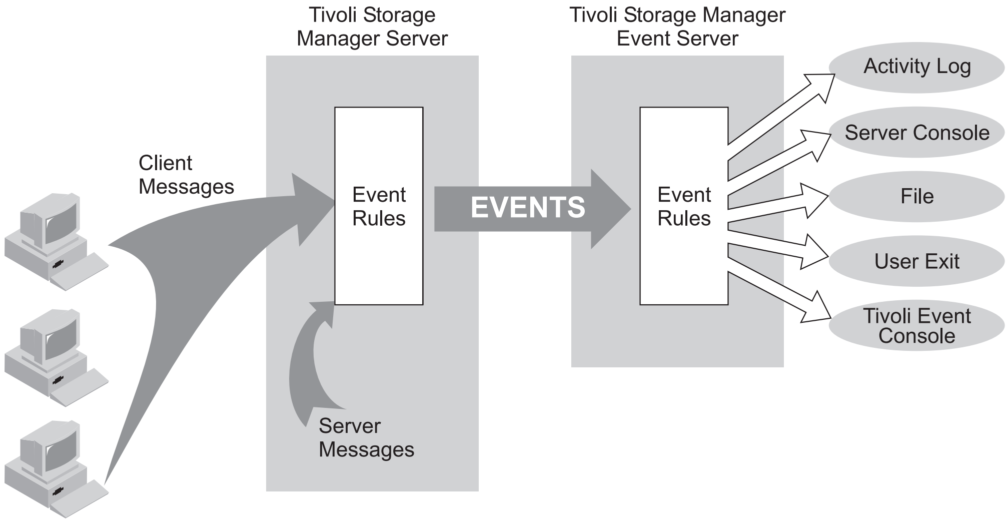


Figure 66. Server to Server Event Logging

The following scenario is a simple example of how enterprise event logging can work.

The administrator at each sending server does the following:

1. Defines the server that will be the event server. For details about communication set up, see "Setting Up Communications for Enterprise Configuration and Enterprise Event Logging" on page 436.

   ```
   define server server_b password=cholla hladdress=9.115.3.45 lladdress=1505
   ```

2. Identifies the server just defined as the event server:

   ```
   define eventserver server_b
   ```

3. Enables the logging of severe, error, and warning server messages from the sending server and severe and error messages from all clients to the event server receiver by issuing the following commands:

   ```
   enable events eventserver severe,error,warning
   enable events eventserver severe,error nodename=*
   ```

4. Begins event logging by issuing the following command:

   ```
   begin eventlogging eventserver
   ```

The administrator at the event server does the following:

5. Enables the logging of severe and error messages to a file named *events* that are sent to it from the sending servers. The administrator defines the file with the following option in the server options file:

   ```
   fileexit yes events append
   ```

   Then the administrator enables the events by issuing the ENABLE EVENTS command for each sending server. For example, for SERVER_A the administrator would enter:

   ```
   enable events file severe,error servername=server_a
   ```

   Note: By default, logging of events from another server is enabled to the event server activity log. However, unlike events originating from a local server, events originating from another server can be disabled for the activity log at an event server.

One or more servers can send events to an event server. An administrator at the event server enables the logging of specific events from specific servers. In the previous example, SERVER_A routes severe, error, and warning messages to SERVER_B. SERVER_B, however, logs only the severe and error messages. If a third server sends events to SERVER_B, logging is enabled only if an ENABLE EVENTS command includes the third server. Furthermore, the SERVER_B determines the receiver to which the events are logged.

Attention: It is important that you do not set up server-to-server event logging in a loop. In such a situation, an event would continue logging indefinitely, tying up network and memory resources. TSM will detect such a situation and issue a message. Here are a few configurations to avoid:

- SERVER_A logs to SERVER_B, and SERVER_B logs to SERVER_A.
- SERVER_A logs to SERVER_B; SERVER_B logs to SERVER_C; SERVER_C logs to SERVER_A.

Querying Event Logging

The QUERY ENABLED command displays a list of server or client events that are enabled or disabled by a specified receiver. Because the lists of enabled and disabled events could be very long, TSM displays the shorter of the two lists. For example, assume that 1000 events for client node HSTANFORD were enabled for logging to the user exit and that later two events were disabled. To query the enabled events for HSTANFORD, enter:

```
query enabled userexit nodename=hstanford
```

The output would specify the *number* of enabled events and the *message names* of disabled events:

```
998 events are enabled for node HSTANFORD for the USEREXIT receiver.
The following events are DISABLED for the node HSTANFORD for the USEREXIT
receiver:
 ANE4000, ANE49999
```

The QUERY EVENTRULES command displays the history of events that are enabled or disabled by a specific receiver for the server or for a client node.

```
query enabled userexit nodename=hstanford
```

Using Tivoli Decision Support

Tivoli Decision Support (TDS) for Storage Management Analysis is a separate program product that works with Tivoli Storage Manager to let you strategically manage your enterprise network. Storage Management Analysis helps you make decisions concerning storage management by providing an overview of your system performance and resource usage. Using data collected from TSM servers, Storage Management Analysis displays multidimensional views and detailed reports.

Storage Management Analysis requires the Tivoli Storage Management Decision Support Loader, which runs daily to collect and aggregate the data into the database table. See “Scheduling the Decision Support Loader with Tivoli Storage Manager” for information about automating the scheduling of Decision Support Loader runs. The Tivoli Discovery Administrator interface is used to define queries that extract data from the TDS database into a file. Cognos Transformer builds a cube from the file, and Cognos PowerPlay generates reports from the cube. You can use the Tivoli Discovery Interface to view these reports. You can also use Crystal Reports to generate text-based views.

To use Storage Management Analysis on your TSM server or servers, you must first enable event logging of client events to the activity log. See “Logging Events to the Tivoli Storage Manager Server Console and Activity Log” on page 415 for details.

For documentation about TDS for Storage Management Analysis visit the Web site at http://www.tivoli.com/support/storage_mgr/tivolimain.html

Scheduling the Decision Support Loader with Tivoli Storage Manager

You can schedule the Decision Support Loader (DSL) to run automatically using the TSM Scheduler. Before defining a schedule, ensure that the backup-archive client is installed on a dedicated Windows workstation where the DSL is installed.

Use the following procedure to schedule the DSL:

1. **On the TSM server:**
   a. Register the client node. Assume that the client node you registered is called ASTROdsl.

b. Define a client schedule on the server from which the DSL will extract data. For example, if the schedule is called TSMDSL and client node ASTROdsl is registered in the STANDARD domain, enter:

```
define schedule standard tsm_dsl action=c
object='"c:\program files\tivoli\tsm\decision\tsmdsl"'
```

Notes:

1) The installation directory path for the DSL is:

```
"c:\program files\tivoli\tsm\decision\tsmdsl.exe"
```

2) Enclose the full directory path in quotation marks as shown in the previous example.

c. Associate the client node to the *tsm_dsl* schedule. For example:

```
define association standard tsm_dsl ASTROdsl
```

2. **On the client's workstation:**

a. Ensure that the scheduler is installed.

b. Start the scheduler for the client. Leave the scheduler running until scheduled rollups are no longer needed. To start the scheduler, you can open a command prompt window and navigate to where the backup-archive client is installed and enter:

```
> dsmc schedule
```

Note: If the DSL is not processed according to the schedule you have defined, check the directory path where the DSL is installed.

Monitoring Tivoli Storage Manager Accounting Records

Task	Required Privilege Class
Set accounting records on or off	System

TSM accounting records show the server resources that are used during a session. This information lets you track resources that are used by a client node session. At installation, accounting defaults to OFF. You can set accounting to ON by entering:

```
set accounting on
```

When accounting is on, the server creates a session resource usage accounting record whenever a client node session ends.

Accounting records are stored in the *dsmaccnt.log* file. The DSMSERV_ACCOUNTING_DIR environment variable specifies the directory where the accounting file is opened. If this variable is not set when the server is started, the *dsmaccnt.log* file is placed in the current directory when the server starts. For example, to set the environment variable to place the accounting records in the */home/engineering* directory, enter this command:

```
export DSMSERV_ACCOUNTING_DIR=/home/engineering
```

The accounting file contains text records that can be viewed directly or can be read into a spreadsheet program. The file remains opened while the server is running and accounting is set to ON. The file continues to grow until you delete it or prune old records from it. To close the file for pruning, either temporarily set accounting off or stop the server.

There are 31 fields, which are delimited by commas (,). Each record ends with a new-line character. Each record contains the following information:

Field	Contents
1	Product version
2	Product sublevel
3	Product name, ‘ADSM’,
4	Date of accounting (mm/dd/yyyy)
5	Time of accounting (hh:mm:ss)
6	Node name of TSM client
7	Client owner name (UNIX)
8	Client Platform
9	Authentication method used
10	Communication method used for the session
11	Normal server termination indicator (Normal=X'01', Abnormal=X'00')
12	Number of archive store transactions requested during the session
13	Amount of archived files, in kilobytes, sent by the client to the server
14	Number of archive retrieve transactions requested during the session
15	Amount of space, in kilobytes, retrieved by archived objects
16	Number of backup store transactions requested during the session
17	Amount of backup files, in kilobytes, sent by the client to the server
18	Number of backup retrieve transactions requested during the session
19	Amount of space, in kilobytes, retrieved by backed up objects
20	Amount of data, in kilobytes, communicated between the client node and the server during the session
21	Duration of the session, in seconds
22	Amount of idle wait time during the session, in seconds
23	Amount of communications wait time during the session, in seconds
24	Amount of media wait time during the session, in seconds
25	Client session type. A value of 1 or 4 indicates a general client session. A value of 5 indicates a client session that is running a schedule.
26	Number of space-managed store transactions requested during the session
27	Amount of space-managed data, in kilobytes, sent by the client to the server
28	Number of space-managed retrieve transactions requested during the session
29	Amount of space, in kilobytes, retrieved by space-managed objects
30	Product release
31	Product level

The following shows a sample record:

```
3,8,ADSM,08/03/2000,16:26:37,node1,,AIX,1,Tcp/Ip,0,254,1713,0,0,47,1476,0,0,3316,960,27,5,1,4,0,0,0,0,7,2
```

Daily Monitoring Scenario

This section contains an example of the daily monitoring of a Tivoli Storage Manager system. Depending on the configuration of your system, you may want to perform additional monitoring tasks. If a function does not complete properly, you can review the activity log for errors that occurred at about the time of failure (see “Requesting Information from the Activity Log” on page 412 for details).

You can include the commands shown in a command script that you can run daily. Review the output of the script for any errors or problems.

1. Verify that drives are online. If there is a drive in the unavailable state, there may be errors with schedules.

   ```
   query drive
   ```

2. Verify that database and recovery log volumes are online and synchronized.

   ```
   query dbvolume
   query logvolume
   ```

3. Check the status of disk volumes. If any are offline, check for hardware problems.

   ```
   query volume devclass=disk
   ```

4. Check that scratch volumes are available.

   ```
   query libvolume
   ```

5. Check the access state of the tape volumes. For example, a volume that is not in the read-write state may indicate a problem. You may need to move data and check the volumes out of the library.

   ```
   query volume
   ```

6. Check database and recovery log statistics.

   ```
   query db
   query log
   ```

7. Verify that scheduled database backups completed successfully.

   ```
   query volhistory type=dbbackup
   ```

8. Check the activity log for error messages.

   ```
   query actlog search=ANR????E
   ```

Chapter 20. Working with a Network of Tivoli Storage Manager Servers

You may have a number of TSM servers in your network, at the same or different locations. TSM provides functions to help you configure, manage, and monitor the servers connected to a network. An administrator working at one TSM server can work with TSM servers at other locations around the world.

See the following sections:

Concepts:
"Concepts for Working with a Network of Servers"
Tasks:
"Planning for Enterprise Administration" on page 435
"Setting Up Communications Among Servers" on page 436
"Setting Up an Enterprise Configuration" on page 443
"Performing Tasks on Multiple Servers" on page 464
"Using Virtual Volumes to Store Data on Another Server" on page 469

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Concepts for Working with a Network of Servers

To manage a network of servers, you can use the following capabilities of Tivoli Storage Manager:

- Configure and manage multiple servers with enterprise configuration.

  Distribute a consistent configuration for TSM servers through a configuration manager to managed servers. By having consistent configurations, you can simplify the management of a large number of servers and clients.
- Perform tasks on multiple servers by using command routing, enterprise logon, and enterprise console .
- Send server and client events to another server for logging.
- Monitor many servers and clients from a single server.
- Store data on another server using virtual volumes.

In a network of TSM servers, a server can play a number of different roles. For example, a server may send volumes to be archived on another server and also receive routed commands from another server. In the following descriptions, when a server sends data it is sometimes referred to as a *source server*, and when a server receives data it is sometimes referred to as a *target server*. In other words, one TSM

© Copyright IBM Corp. 1993, 2002

server may be both a source and a target server. At the same time, any TSM server can still provide backup, archive, and space management services to clients.

Configuring and Managing Servers: Enterprise Configuration

The enterprise configuration functions of the Tivoli Storage Manager make it easier to consistently set up and manage a network of TSM servers. You set up configurations on one server and distribute the configurations to the other servers. You can make changes to configurations and have the changes automatically distributed.

Figure 67 on page 433 illustrates a simple configuration. To use enterprise configuration, you first select the TSM server that is to act as the *configuration manager*. You may want to dedicate a new server for this purpose. At the configuration manager, you define the details of the server configurations that you want to distribute. For example:

- You set up backup and archive policies and client option sets
- You designate one or more administrators to have access to the servers, and control their authority levels
- You define the servers that you want the configuration manager to manage or communicate with, and you set up communications among the servers

In one or more *profiles*, you point to the definitions of the configuration information that you want to use to manage other servers.

On each server that is to receive the configuration information, you identify the server as a *managed server* by defining a *subscription* to one or more profiles owned by the configuration manager. All the definitions associated with the profiles are then copied into the managed server's database. Things defined to the managed server in this way are managed objects that cannot be changed by the managed server. From then on, the managed server gets any changes to the managed objects from the configuration manager via the profiles. Managed servers receive changes to configuration information at time intervals set by the servers, or by command.

See "Setting Up an Enterprise Configuration" on page 443 for details.

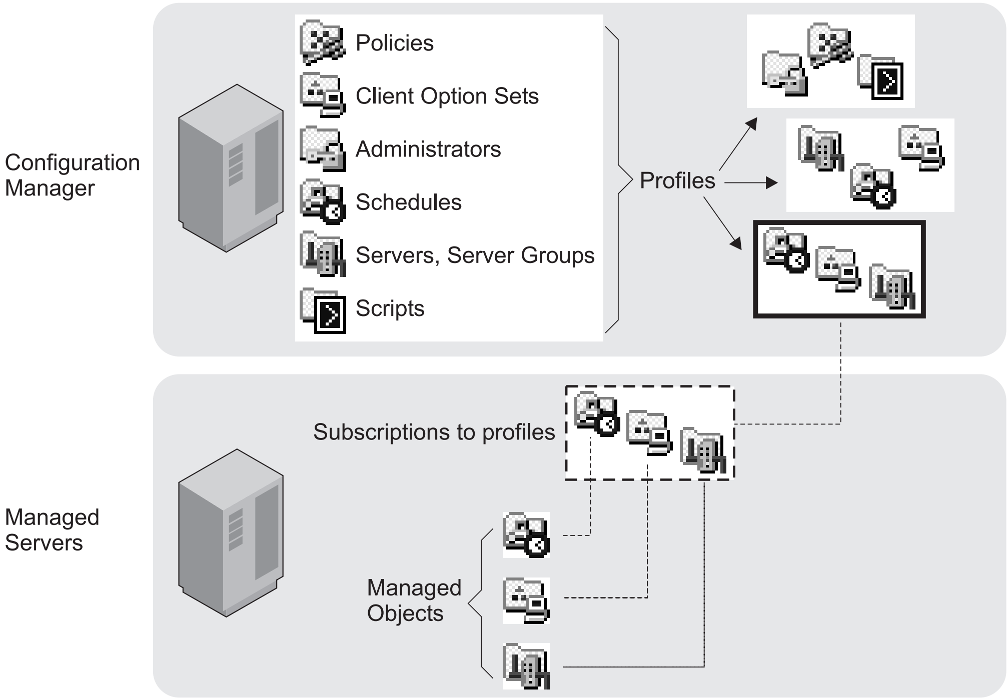


Figure 67. Enterprise Configuration

Performing Tasks on Multiple Servers

When you connect to the configuration manager via a Web browser, you are presented with the *enterprise console* . From the enterprise console you can perform tasks on the configuration manager and on one or more of the managed servers. You can also connect to another server to perform tasks directly on that server. As long as you are registered with the same administrator ID and password, you can do this work on many servers without having to log on each time. See “Using Tivoli Storage Manager Enterprise Logon” on page 464.

From the command line of the administrative Web interface or from the command-line administrative client, you can also route commands to other servers. The other servers must be defined to the server to which you are connected. You must also be registered on the other servers as an administrator with the administrative authority that is required for the command. See “Routing Commands” on page 465.

To make routing commands easier, you can define a server group that has servers as members. See “Setting Up Server Groups” on page 467. Commands that you route to a server group are sent to all servers in the group.

Central Monitoring

TSM provides you with several ways to centrally monitor the activities of a server network:

- Enterprise event logging, in which events are sent from one or more of servers to be logged at an event server. See “Enterprise Event Logging: Logging Events to Another Server” on page 423 for a description of the function and “Setting Up Communications for Enterprise Configuration and Enterprise Event Logging” on page 436 for communications set up.

- Allowing designated administrators to log in to any of the servers in the network with a single user ID and password. See "Using Tivoli Storage Manager Enterprise Logon" on page 464.
- Routing query commands to one or more of the servers in the network. See "Routing Commands to One or More Servers" on page 466 for a description of the function and "Setting Up Communications for Enterprise Configuration and Enterprise Event Logging" on page 436 for communications set up.

Storing Data on Another Server

TSM lets one server store data in and retrieve data from the storage pool of another server. This data, stored as *virtual volumes*, can include database and storage pool backups, disaster recovery plan files, and data that is directly backed up, archived, or space managed from client nodes. The data can also be a recovery plan file created by using Tivoli Disaster Recovery Manager (DRM). The source server is a client of the target server, and the data for the source server is managed only by the source server. In other words, the source server controls the expiration and deletion of the files that comprise the virtual volumes on the target server.

To use virtual volumes to store database and storage pool backups and recovery plan files, you must have the Tivoli Disaster Recovery Manager product and register a license for its use. See "Licensing Tivoli Storage Manager" on page 349.

For information on using virtual volumes with DRM, see "Chapter 23. Using Tivoli Disaster Recovery Manager" on page 545.

Example Scenarios

The functions for managing multiple servers can be applied in many ways. Here are just two scenarios to give you some ideas about how you can put the functions to work for you:

- Setting up and managing TSM servers primarily from one location. For example, an administrator at one location controls and monitors servers at several locations.
- Setting up a group of TSM servers from one location, and then managing the servers from any of the servers. For example, several administrators are responsible for maintaining a group of servers. One administrator defines the configuration information on one server for distributing to servers in the network. Administrators on the individual servers in the network manage and monitor the servers.

Managing Tivoli Storage Manager Servers from One Location

Enterprise management allows you to set up and manage the servers in your network from one location, the enterprise console . For example, suppose you are an administrator responsible for TSM servers at your own location plus servers at branch office locations. Servers at each location have similar storage resources and client requirements. You can set up the environment as follows:

- Set up an existing or new TSM server as a configuration manager.
- Set up communications so that a configuration manager can send commands to its managed servers.
- Define the configuration you want to distribute by defining policy domains, schedules, and so on. Associate the configuration information with profiles.
- Have the managed servers subscribe to profiles.
- Activate policies and set up storage pools as needed on the managed servers.

- Set up enterprise monitoring by setting up one server as an event server. The event server can be the same server as the configuration manager or a different server.

After you complete the setup, you can manage many servers as if there was just one. You can do any of the following tasks:

- Have administrators that can manage the group of servers from anywhere in the network by using the enterprise console, an interface available through a Web browser.
- Have consistent policies, schedules, and client option sets on all servers.
- Make changes to configurations and have the changes automatically distributed to all servers. Allow local administrators to monitor and tune their own servers.
- Perform tasks on any server or all servers by using command routing from the enterprise console.
- Back up the databases of the managed servers on the automated tape library that is attached to the server that is the configuration manager. You use virtual volumes to accomplish this.
- Log on to individual servers from the enterprise console without having to re-enter your password, if your administrator ID and password are the same on each server.

Managing Servers from Any Server

Enterprise management allows you to manage the servers in your network from many locations. For example, suppose you are an administrator responsible for servers located in different departments on a college campus. The servers have some requirements in common, but also have many unique client requirements. You can set up the environment as follows:

- Set up an existing or new TSM server as a configuration manager.
- Set up communications so that commands can be sent from any server to any other server.
- Define any configuration that you want to distribute by defining policy domains, schedules, and so on, on the configuration manager. Associate the configuration information with profiles.
- Have the managed servers subscribe to profiles as needed.
- Activate policies and set up storage pools as needed on the managed servers.
- Set up enterprise monitoring by setting up one server as an event server. The event server can be the same server as the configuration manager or a different server.

After setting up in this way, you can manage the servers from any server. You can do any of the following tasks:

- Use enterprise console to monitor all the servers in your network.
- Perform tasks on any or all servers using the enterprise console and command routing.
- Manage the group of servers from anywhere in the network. Allow local administrators to monitor and tune their own servers.

Planning for Enterprise Administration

To take full advantage of the functions of Enterprise Administration, you should decide on the following:

- The servers you want to include in the enterprise network. The servers must have unique names.
- The server or servers from which you want to manage the network. Servers can have multiple roles. For example, one server can act as a server for backup-archive clients, as the configuration manager, and as the event server. You can also set up separate servers to fill each of these roles.
- Whether you want administrators to have the ability to route commands to other servers. If you want administrators to route commands, decide on the servers from which and to which commands will be routed.
- The administrator activities you want to be centrally managed.
- The authority level of the administrators and the servers to which they should have access.

Setting Up Communications Among Servers

This section describes how to set up communications for enterprise configuration, enterprise event logging, and command routing. Communication setup for server-to-server virtual volumes is described in "Setting Up Source and Target Servers for Virtual Volumes" on page 471.

When you set up communications among servers for any purpose, ensure that servers have unique names. At installation, a TSM server has the name "TSM". Change the server name to a unique name before setting up communication with other servers. For example, enter this command to name the server TUCSON:

```
set servername tucson
```

Setting Up Communications for Enterprise Configuration and Enterprise Event Logging

The communication setup for enterprise configuration and enterprise event logging, which is through TCP/IP, is identical. The examples shown here apply to both functions. If you are set up for one, you are set up for the other. However, be aware that the configuration manager and event server are not defined simply by setting up communications. You must identify a server as a configuration manager (SET CONFIGMANAGER command) or an event server (DEFINE EVENTSERVER command). Furthermore, a configuration manager and an event server can be the same server or different servers.

Enterprise configuration
Each managed server must be defined to the configuration manager, and the configuration manager must be defined to each managed server.

Enterprise event logging
Each server sending events to an event server must be defined to the event server, and the event server must be defined to each source server.

The following examples of setting up communications could be used to create these configurations:

- A server named HEADQUARTERS as a configuration manager and two servers, MUNICH and STRASBOURG, as managed servers.
- HEADQUARTERS as an event server and MUNICH and STRASBOURG as source servers.

For a pair of servers to communicate with each other, each server must be defined to the other. For example, if a configuration manager manages three managed servers, there are three server pairs. You can issue separate definitions from each server in each pair, or you can “cross define” a pair in a single operation. Cross definition can be useful in large or complex networks. The following scenarios and accompanying figures illustrate the two methods.

Using separate definitions — Follow this sequence:

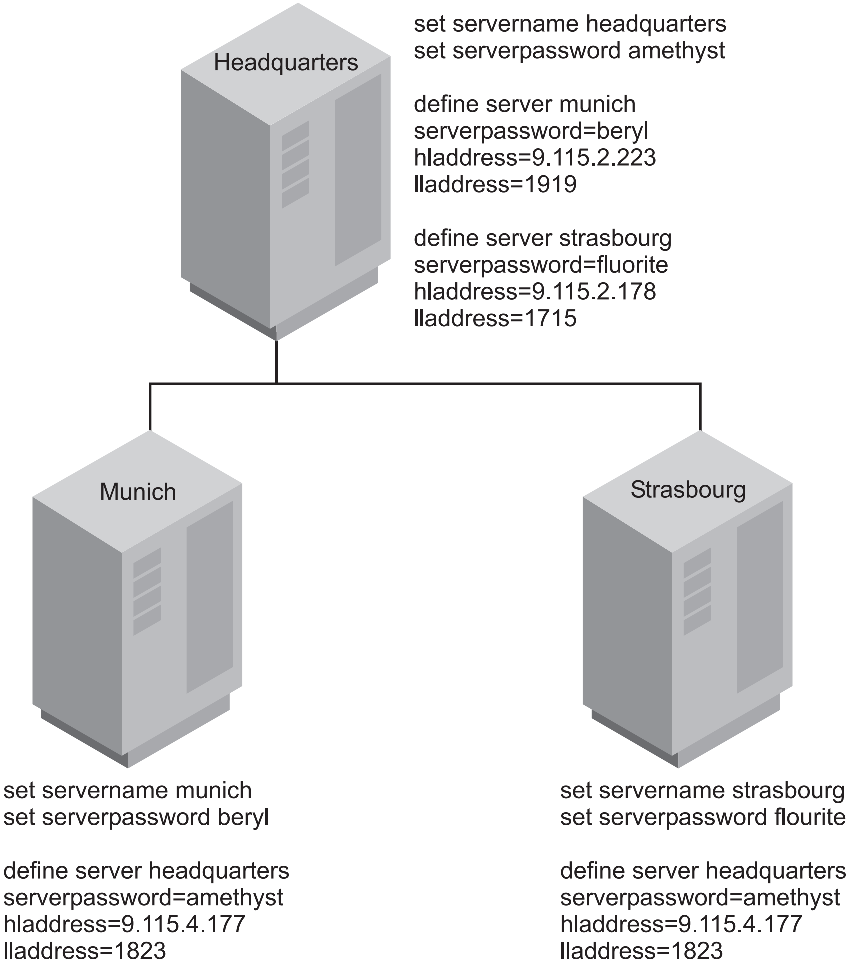
1. **On MUNICH**: Specify the server name and password of MUNICH.

   On STRASBOURG: Specify the server name and password of STRASBOURG.

   On HEADQUARTERS: Specify the server name and password of HEADQUARTERS.
2. **On HEADQUARTERS**: Define MUNICH (whose password is BERYL and whose address is 9.115.2.223:1919) and STRASBOURG (whose password is FLUORITE and whose address is 9.115.2.178:1715).

   On MUNICH and STRASBOURG: Define HEADQUARTERS (whose password is AMETHYST and whose address is 9.115.4.177:1823).

Figure 68 shows the servers and the commands issued on each:



```
set servername headquarters
set serverpassword amethyst

define server munich
serverpassword=beryl
hladdress=9.115.2.223
lladdress=1919

define server strasbourg
serverpassword=fluorite
hladdress=9.115.2.178
lladdress=1715
```

```
set servername munich
set serverpassword beryl

define server headquarters
serverpassword=amethyst
hladdress=9.115.4.177
lladdress=1823
```

```
set servername strasbourg
set serverpassword flourite

define server headquarters
serverpassword=amethyst
hladdress=9.115.4.177
lladdress=1823
```

Figure 68. Communication Configuration with Separate Server Definitions

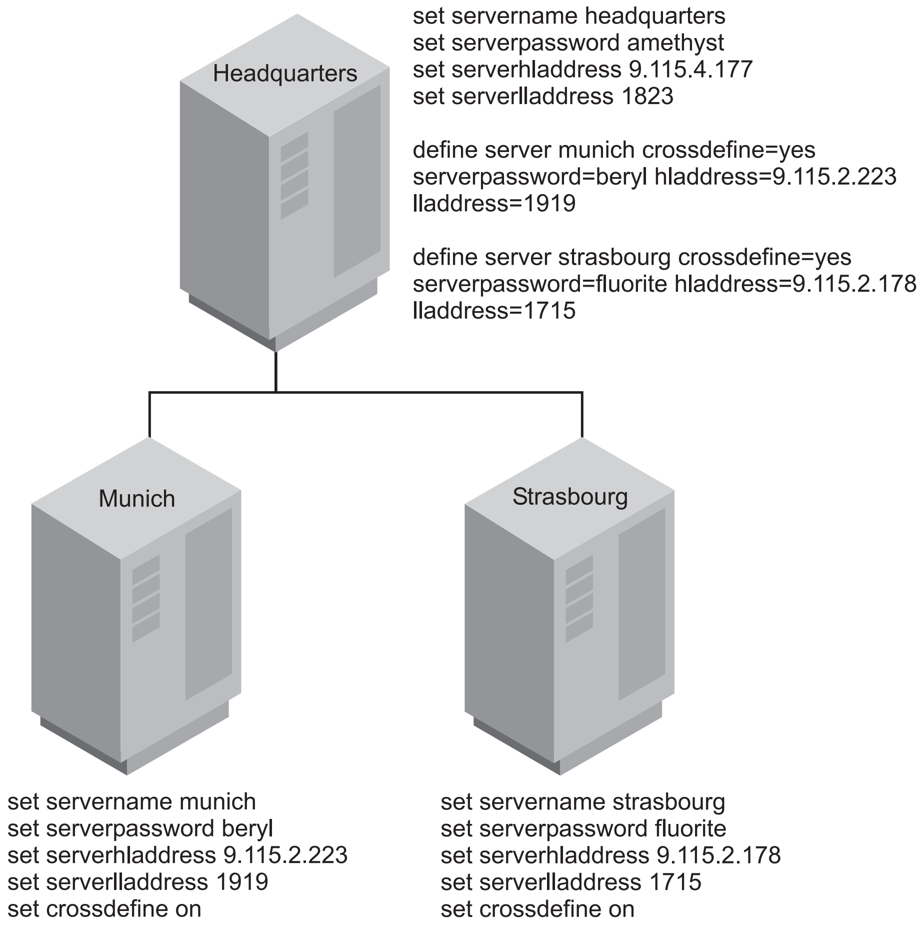
Using Cross Definitions — Follow this sequence:

1. **On MUNICH**: Specify the server name, password, and high and low level addresses of MUNICH. Specify that cross define is permitted.

   On STRASBOURG: Specify the server name, password, and high and low level addresses of STRASBOURG. Specify that cross define is permitted.

   On HEADQUARTERS: Specify the server name, password, and high and low level addresses of HEADQUARTERS.
2. **On HEADQUARTERS**: Define MUNICH and STRASBOURG, specifying that cross define should be done.

Figure 69 shows the servers and the commands issued on each:



```
set servername headquarters
set serverpassword amethyst
set serverhladdress 9.115.4.177
set serverlladdress 1823

define server munich crossdefine=yes
serverpassword=beryl hladdress=9.115.2.223
lladdress=1919

define server strasbourg crossdefine=yes
serverpassword=fluorite hladdress=9.115.2.178
lladdress=1715
```

```
set servername munich
set serverpassword beryl
set serverhladdress 9.115.2.223
set serverlladdress 1919
set crossdefine on
```

```
set servername strasbourg
set serverpassword fluorite
set serverhladdress 9.115.2.178
set serverlladdress 1715
set crossdefine on
```

Figure 69. Communication Configuration with Cross Definition

Communication Security

Security for this communication configuration is enforced through the exchange of passwords (which are encrypted) and, in the case of enterprise configuration only, verification keys. Communication among servers, which is through TCP/IP, requires that the servers verify server passwords (and verification keys). For example, assume that HEADQUARTERS begins a session with MUNICH:

1. HEADQUARTERS, the source server, identifies itself by sending its name to MUNICH.
2. The two servers exchange verification keys (enterprise configuration only).
3. HEADQUARTERS sends its password to MUNICH, which verifies it against the password stored in its database.

4. If MUNICH verifies the password, it sends its password to HEADQUARTERS, which, in turn, performs password verification.

Note: If another server named MUNICH tries to contact HEADQUARTERS for enterprise configuration, the attempt will fail. This is because the verification key will not match. If MUNICH was moved or restored, you can issue the UPDATE SERVER command with the FORCERESYNC parameter to override the condition.

Setting Up Communications for Command Routing

This section describes how to set up communications for command routing. You must define the target servers to the source servers, and the same administrator must be registered on all servers. Using enterprise configuration, you can easily distribute the administrator information to all the servers.

Note: You must be registered as an administrator with the same name and password on the source server and all target servers. The privilege classes do not need to be the same on all servers. However, to successfully route a command to another server, an administrator must have the minimum required privilege class for that command on the server from which the command is being issued.

For command routing in which one server will always be the sender, you would only define the target servers to the source server. If commands can be routed from any server to any other server, each server must be defined to all the others.

Only One Source Server

The example in this section shows how to set up communications for administrator HQ on the server HEADQUARTERS who will route commands to the servers MUNICH and STRASBOURG. Administrator HQ has the password SECRET and has system privilege class. Here is the procedure:

- **On HEADQUARTERS**: register administrator HQ and specify the server names and addresses of MUNICH and STRASBOURG:

```
register admin hq secret
grant authority hq classes=system

define server munich hladdress=9.115.2.223 lladdress=1919
define server strasbourg hladdress=9.115.2.178 lladdress=1715
```

- **On MUNICH and STRASBOURG** Register administrator HQ with the required privilege class on each server:

```
register admin hq secret
grant authority hq classes=system
```

Note: If your server network is using enterprise configuration, you can automate the preceding operations. You can distribute the administrator and server lists to MUNICH and STRASBOURG. In addition, all server definitions and server groups are distributed by default to a managed server when it first subscribes to any profile on a configuration manager. Therefore, it receives all the server definitions that exist on the configuration manager, thus enabling command routing among the servers.

Multiple Source Servers

The examples in this section show how to set up communications if the administrator, HQ, can route commands from any of the three servers to any of the other servers. You must define all the servers to each other. You can separately define each server to each of the other servers, or you can “cross define” the servers. In cross definition, defining MUNICH to HEADQUARTERS also results in automatically defining HEADQUARTERS to MUNICH.

Separate Definitions: Follow this sequence:

1. **On MUNICH:** Specify the server name and password of MUNICH. Register administrator HQ and grant HQ system authority.

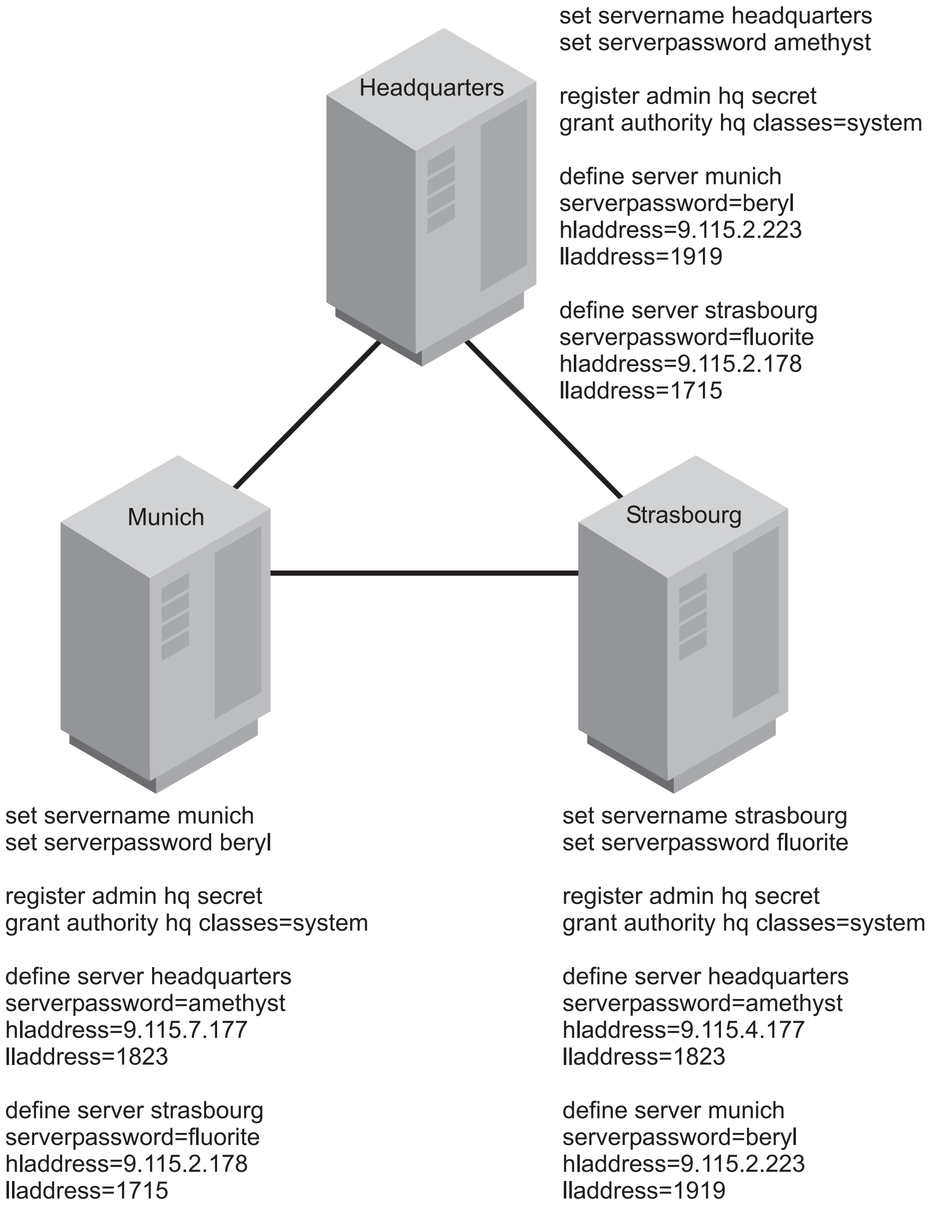
   On STRASBOURG: Specify the server name and password of STRASBOURG. Register administrator HQ and grant HQ system authority.

   On HEADQUARTERS: Specify the server name and password of HEADQUARTERS. Register administrator HQ and grant HQ system authority.
2. **On HEADQUARTERS:** Define MUNICH (whose password is BERYL and whose address is 9.115.2.223:1919) and STRASBOURG (whose password is FLUORITE and whose address is 9.115.2.178:1715).

   On MUNICH: Define HEADQUARTERS (whose password is AMETHYST and whose address is 9.115.4.177:1823) and STRASBOURG.

   On STRASBOURG: Define HEADQUARTERS and MUNICH.

Figure 70 on page 441 shows the servers and the commands issued on each:



```
set servername headquarters
set serverpassword amethyst

register admin hq secret
grant authority hq classes=system

define server munich
serverpassword=beryl
hladdress=9.115.2.223
lladdress=1919

define server strasbourg
serverpassword=fluorite
hladdress=9.115.2.178
lladdress=1715
```

```
set servername munich
set serverpassword beryl

register admin hq secret
grant authority hq classes=system

define server headquarters
serverpassword=amethyst
hladdress=9.115.7.177
lladdress=1823

define server strasbourg
serverpassword=fluorite
hladdress=9.115.2.178
lladdress=1715
```

```
set servername strasbourg
set serverpassword fluorite

register admin hq secret
grant authority hq classes=system

define server headquarters
serverpassword=amethyst
hladdress=9.115.4.177
lladdress=1823

define server munich
serverpassword=beryl
hladdress=9.115.2.223
lladdress=1919
```

Figure 70. Communication Configuration with Separate Server Definitions

Cross Definitions: Follow this sequence:

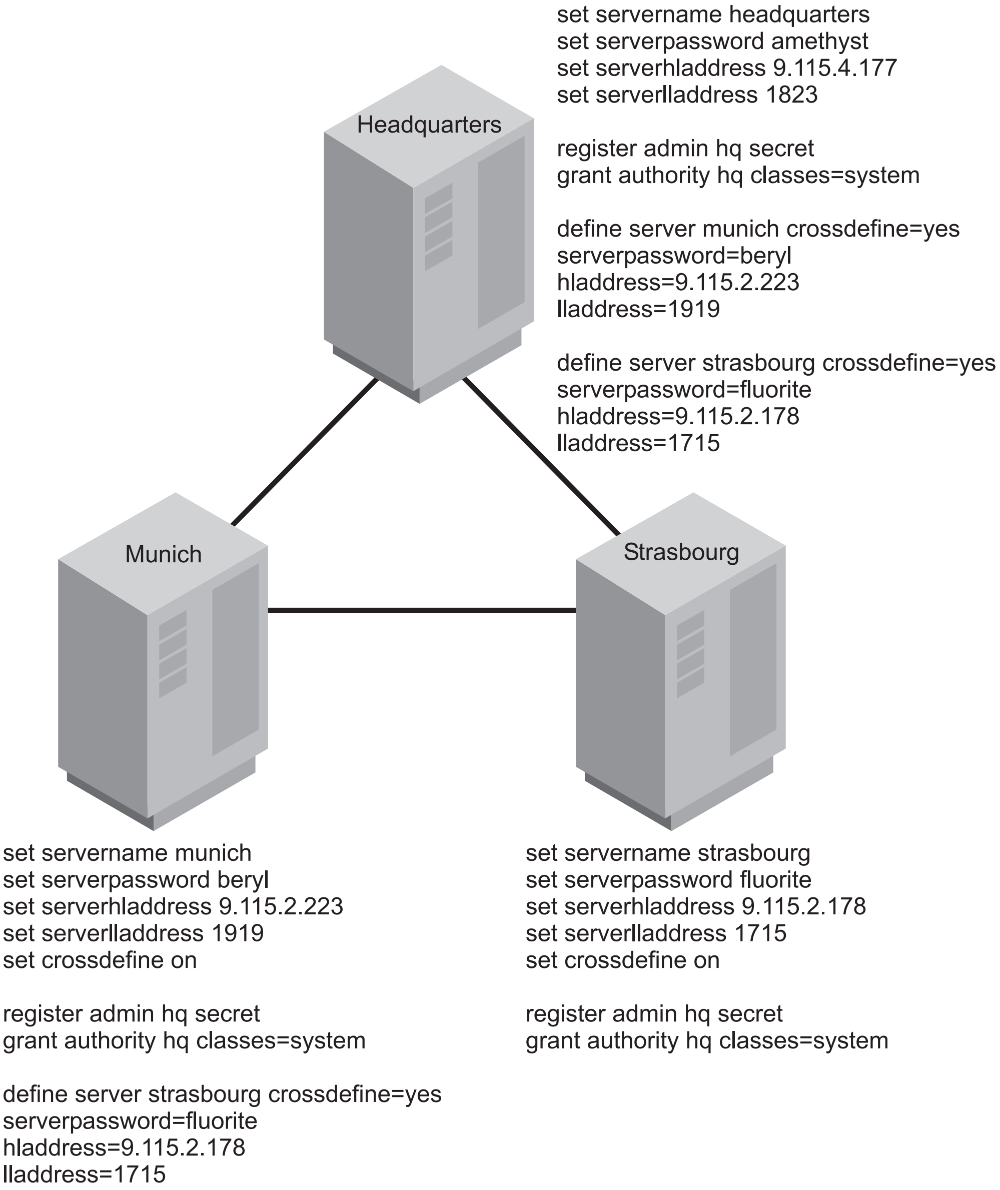
1. **On MUNICH:** Specify the server name, password, and high and low level addresses of MUNICH. Specify that cross define is permitted. Register administrator HQ and grant HQ system authority.

   On STRASBOURG: Specify the server name, password, and high and low level addresses of STRASBOURG. Specify that cross define is permitted. Register administrator HQ and grant HQ system authority.

   On HEADQUARTERS: Specify the server name, password, and high and low level addresses of HEADQUARTERS. Register administrator HQ and grant HQ system authority.
2. **On HEADQUARTERS:** Define MUNICH and STRASBOURG, specifying that cross define should be done.
3. **On MUNICH:** Define STRASBOURG, specifying that cross define should be done.

Note: If your server network is using enterprise configuration, you can automate the preceding operations. You can distribute the administrator lists and server lists to MUNICH and STRASBOURG. In addition, all server definitions and server groups are distributed by default to a managed server when it first subscribes to any profile on a configuration manager. Therefore, it receives all the server definitions that exist on the configuration manager, thus enabling command routing among the servers.

Figure 71 shows the servers and the commands issued on each:



```
set servername headquarters
set serverpassword amethyst
set serverhladdress 9.115.4.177
set serverlladdress 1823

register admin hq secret
grant authority hq classes=system

define server munich crossdefine=yes
serverpassword=beryl
hladdress=9.115.2.223
lladdress=1919

define server strasbourg crossdefine=yes
serverpassword=fluorite
hladdress=9.115.2.178
lladdress=1715
```

```
set servername munich
set serverpassword beryl
set serverhladdress 9.115.2.223
set serverlladdress 1919
set crossdefine on

register admin hq secret
grant authority hq classes=system

define server strasbourg crossdefine=yes
serverpassword=fluorite
hladdress=9.115.2.178
lladdress=1715
```

```
set servername strasbourg
set serverpassword fluorite
set serverhladdress 9.115.2.178
set serverlladdress 1715
set crossdefine on

register admin hq secret
grant authority hq classes=system
```

Figure 71. Communication Configuration with Cross Definitions

Updating and Deleting Servers

You can update a server definition by issuing the UPDATE SERVER command.

- For Server-to-server Virtual Volumes:
  - If you update the node name, you must also update the password.

  - If you update the password but not the node name, the node name defaults to the server name specified by the SET SERVERNAME command.
- For enterprise configuration and enterprise event logging: If you update the server password, it must match the password specified by the SET SERVERPASSWORD command at the target server.
- For enterprise configuration: When a server is first defined at a managed server, that definition cannot be replaced by a server definition from a configuration manager. This prevents the definition at the managed server from being inadvertently replaced. Such a replacement could disrupt functions that require communication among servers, for example command routing or virtual volumes.

  To allow replacement, update the definition at the managed server by issuing the UPDATE SERVER command with the ALLOWREPLACE=YES parameter. When a configuration manager distributes a server definition, the definition always includes the ALLOWREPLACE=YES parameter.

You can delete a server definition by issuing the DELETE SERVER command. For example, to delete the server named NEWYORK, enter the following:

```
delete server newyork
```

The deleted server is also deleted from any server groups of which it is a member. See "Setting Up Server Groups" on page 467 for information about server groups.

You cannot delete a server if either of the following conditions is true:

- The server is defined as an event server.

  You must first issue the DELETE EVENTSERVER command.
- The server is a target server for virtual volumes.

  A target server is named in a DEFINE DEVCLASS (DEVTYPE=SERVER) command. You must first change the server name in the device class or delete the device class.

Setting Up an Enterprise Configuration

After you set up server communication as described in "Setting Up Communications for Enterprise Configuration and Enterprise Event Logging" on page 436, you set up the configuration manager and its profiles. With the profiles, you designate the configuration information that can be distributed to managed servers. Then you can set up other servers as managed servers. The managed servers receive configuration information through subscriptions to profiles on the configuration manager. Each managed server stores the distributed information as managed objects in its database. Managed servers receive periodic updates of the configuration information from the configuration manager, or an administrator can trigger an update by command.

You can distribute the following configuration information from a configuration manager to managed servers:

- Administrators, including authorities for them

  Note: For administrator definitions that have node authority, the configuration manager only distributes information such as password and contact information. Node authority for the managed administrator can be controlled on the managed server using the GRANT AUTHORITY and REVOKE AUTHORITY commands specifying the CLASS=NODE parameter.

- Policy objects

  Policy objects include policy domains, and the policy sets, management classes, copy groups and client schedules associated with them. However, a configuration manager does *not* distribute an active policy set and any of its associated objects. On each managed server, you must activate a policy set in each managed policy domain.

  Note: The configuration manager does not distribute definitions for any storage pools identified as destinations in the policy. Definitions of storage pools and device classes are not distributed by a configuration manager.
- Administrative command schedules
- TSM server scripts
- Client option sets
- Server definitions
- Server groups

"Enterprise Configuration Scenario" gives you an overview of the steps to take for one possible implementation of enterprise configuration. Sections that follow give more details on each step.

Enterprise Configuration Scenario

To illustrate how you might use these functions, suppose your enterprise has offices around the world, with one or more TSM servers at each location. To make managing these servers easier, you want to control the configuration of all TSM servers from one TSM server in the headquarters office. Figure 72 shows the hierarchy that you want to set up.

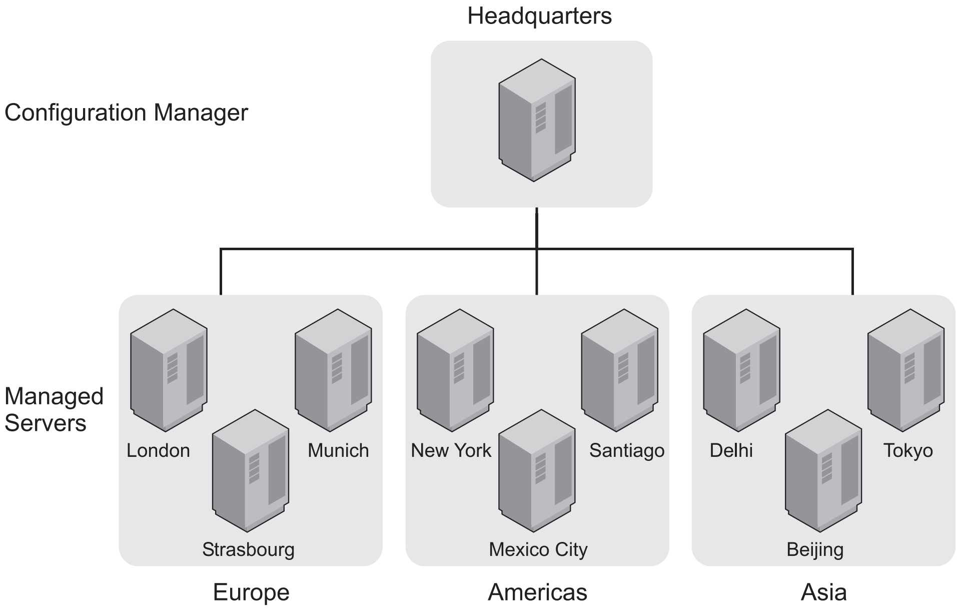


Figure 72. A Scenario for Implementing Enterprise Configuration

You want to set up a configuration manager named HEADQUARTERS. Managed servers have the names of cities where they are located. You have three groups of managed servers, one in the Americas, one in Europe, and one in Asia. Each of the servers supports backup and archive services for client machines in that office. For client backup operations, you want to use the default policy that stores backups on disk. Each server has an automated tape library configured to work with TSM, and you want to use the tape library at each location for client archive operations and for TSM server database backups. You want to be able to monitor activities on all servers. You also want to designate some other users as administrators who can work with these servers.

The following sections give you an overview of the steps to take to complete this setup. For details on each step, see the section referenced.

Setting up a Configuration Manager

Figure 73 shows the specific commands needed to set up one TSM server as a configuration manager. The following procedure gives you an overview of the steps required to set up a server as a configuration manager.

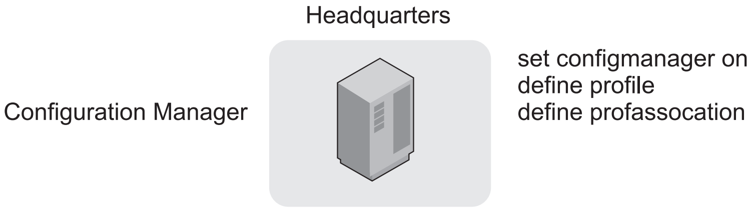


Figure 73. Setting Up a Configuration Manager

1. Decide whether to use the existing TSM server in the headquarters office as the configuration manager or to install a new TSM server on a system.
2. Set up the communications among the servers. See "Setting Up Communications Among Servers" on page 436 for details.
3. Identify the server as a configuration manager.

   Use the following command:

   ```
   set configmanager on
   ```

   This command automatically creates a profile named DEFAULT_PROFILE. The default profile includes all the server and server group definitions on the configuration manager. As you define new servers and server groups, they are also associated with the default profile. For more information, see "Creating the Default Profile on a Configuration Manager" on page 447.
4. Create the configuration to distribute.

   The tasks that might be involved include:

   - Register administrators and grant authorities to those that you want to be able to work with all the servers.
   - Define policy objects and client schedules
   - Define administrative schedules
   - Define TSM server scripts
   - Define client option sets
   - Define servers
   - Define server groups

   Example 1: You need a shorthand way to send commands to different groups of managed servers. You can define server groups. For example, you can define a server group named AMERICAS for the servers in the offices in North America and South America. See "Defining a Server Group and Members of a Server Group" on page 467 for details.

Example 2: You want each managed server to back up its database and storage pools regularly. One way to do this is to set up TSM server scripts and schedules to automatically run these scripts everyday. You can do the following:

- Verify or define server scripts that perform these operations.
- Verify or define administrative command schedules that run these scripts.

Example 3: You want clients to back up data to the default disk storage pool, BACKUPPOOL, on each server. But you want clients to archive data directly to the tape library attached to each server. You can do the following:

- In the policy domain that you will point to in the profile, update the archive copy group so that TAPEPOOL is the name of the destination storage pool.
- On each server that is to be a managed server, ensure that you have a tape storage pool named TAPEPOOL.

  Note: You must set up the storage pool itself (and associated device class) on each managed server, either locally or by using command routing. If a managed server already has a storage pool associated with the automated tape library, you can rename the pool to TAPEPOOL.

Example 4: You want to ensure that client data is consistently backed up and managed on all servers. You want all clients to be able to store three backup versions of their files. You can do the following:

- Verify or define client schedules in the policy domain so that clients are backed up on a consistent schedule.
- In the policy domain that you will point to in the profile, update the backup copy group so that three versions of backups are allowed.
- Define client option sets so that basic settings are consistent for clients as they are added.

5. Define one or more profiles.

   For example, you can define one profile named ALLOFFICES that points to all the configuration information (policy domain, administrators, scripts, and so on). You can also define profiles for each type of information, so that you have one profile that points to policy domains, and another profile that points to administrators, for example.

   For details, see “Creating and Changing Configuration Profiles” on page 448.

Setting Up a Managed Server

Figure 74 shows the specific commands needed to set up one TSM server as a managed server. The following procedure gives you an overview of the steps required to set up a server as a managed server.

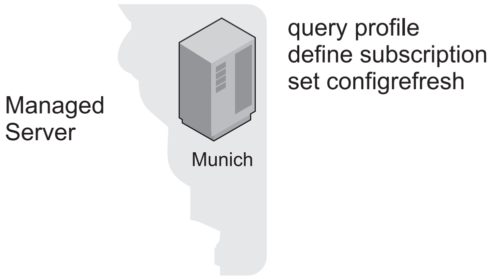


Figure 74. Setting Up a Managed Server

Setting up the managed server can be done by an administrator working at a central location, or by administrators working at the servers that will be managed servers.

A server becomes a managed server when that server first subscribes to a profile on a configuration manager.

1. Query the server to look for potential conflicts.

   See “Getting Information about Profiles” on page 455. Look for definitions of objects on the managed server that have the same name as those defined on the configuration manager. With some exceptions, these objects will be overwritten when the managed server first subscribes to the profile on the configuration manager. See “Associating Configuration Information with a Profile” on page 449 for details on the exceptions.

   If the managed server is a new server and you have not defined anything, the only objects you will find are the defaults (for example, the STANDARD policy domain).
2. Subscribe to one or more profiles.

   A managed server can only subscribe to profiles on one configuration manager. See “Subscribing to a Profile” on page 457.

   If you receive error messages during the configuration refresh, such as a local object that could not be replaced, resolve the conflict and refresh the configuration again. You can either wait for the automatic refresh period to be reached, or kick off a refresh by issuing the SET CONFIGREFRESH command, setting or resetting the interval.
3. If the profile included policy domain information, activate a policy set in the policy domain, add or move clients to the domain, and associate any required schedules with the clients.

   You may receive warning messages about storage pools that do not exist, but that are needed for the active policy set. Define any storage pools needed by the active policy set, or rename existing storage pools. See “Defining or Updating Primary Storage Pools” on page 158 or “Renaming a Storage Pool” on page 218.
4. If the profile included administrative schedules, make the schedules active.

   Administrative schedules are not active when they are distributed by a configuration manager. The schedules do not run on the managed server until you make them active on the managed server. See “Tailoring Schedules” on page 369.
5. Set how often the managed server contacts the configuration manager to update the configuration information associated with the profiles.

   The initial setting for refreshing the configuration information is 60 minutes. See “Refreshing Configuration Information” on page 461.

Creating the Default Profile on a Configuration Manager

Task	Required Privilege Class
Set up a server as a configuration manager	System

To set up one TSM server as the source for configuration information for other servers, you identify the server as a configuration manager. A configuration manager can be an existing TSM server that already provides services to clients, or can be a server dedicated to just providing configuration information to other TSM servers.

Enter the following command:

```
set configmanager on
```

When a server becomes a configuration manager, the server automatically creates a default profile named DEFAULT_PROFILE. The default profile contains any definitions of servers and server groups that exist on the configuration manager. You can change or delete the profile named DEFAULT_PROFILE.

When a managed server first subscribes to a profile on a configuration manager, the configuration manager automatically also subscribes the managed server to the profile named DEFAULT_PROFILE, if it exists. The information distributed via this profile gets refreshed in the same way as other profiles. This helps ensure that all servers have a consistent set of server and server group definitions for all servers in the network.

If you do not change the DEFAULT_PROFILE, whenever a managed server subscribed to the DEFAULT_PROFILE profile refreshes configuration information, the managed server receives definitions for all servers and server groups that exist on the configuration manager at the time of the refresh. As servers and server groups are added, deleted, or changed on the configuration manager, the changed definitions are distributed to subscribing managed servers.

Creating and Changing Configuration Profiles

You create configuration profiles on a configuration manager, which distributes the information associated with the profiles to any managed server that subscribes to those profiles. Creating a configuration profile includes these steps:

1. Defining the profile
2. Associating the configuration information with the profile

Once you define the profile and its associations, a managed server can subscribe to the profile and obtain the configuration information.

After you define a profile and associate information with the profile, you can change the information later. While you make changes, you can lock the profiles to prevent managed servers from refreshing their configuration information. To distribute the changed information associated with a profile, you can unlock the profile, and either wait for each managed server to refresh its configuration to get the changed information or notify each managed server to refresh its configuration. The following sections provide information on each of these tasks.

Defining the Profile

Task	Required Privilege Class
Define profiles	System

When you define the profile, you select the name and can include a description. For example, to define a profile named ALLOFFICES, enter the following command:

```
define profile alloffices
 description='Configuration to be used by all offices'
```

Associating Configuration Information with a Profile

Task	Required Privilege Class
Define profile associations	System

After you define a profile, you associate the configuration information that you want to distribute via that profile. You can associate the following configuration information with a profile:

- TSM administrators, including their authorities.

  Note: Be careful if you are distributing definitions of administrators that have the same name as administrators already defined to managed servers. The configuration refresh overwrites the administrator definition and authority defined on the managed server. If the authority level of an administrator is less on the configuration manager than it was on the managed server, you could have problems with access to the managed server after distributing the administrator definition.

  The configuration manager does not distribute information about whether an administrator is locked (preventing access to the server).

  The administrator with the name SERVER_CONSOLE is never distributed from the configuration manager to a managed server.
- Servers and server groups.

  The DEFAULT_PROFILE that is automatically created on a configuration manager already points to all servers and server groups defined to that server. If you leave the DEFAULT_PROFILE intact, you do not need to include servers or server groups in any other profile. Any servers and server groups that you define later are associated automatically with the default profile and the configuration manager distributes the definitions at the next refresh.

  For a server definition, the following attributes are distributed:
  - Communication method
  - TCP/IP address (high-level address)
  - Port number (low-level address)
  - Server password
  - Server URL
  - The description

  When server definitions are distributed, the attribute for allowing replacement is always set to YES. You can set other attributes, such as the server's node name, on the managed server by updating the server definition.

  A managed server may already have a server defined with the same name as a server associated with the profile. The configuration refresh does not overwrite the local definition unless the managed server allows replacement of that definition. On a managed server, you allow a server definition to be replaced by updating the local definition. For example:

  ```
  update server santiago allowreplace=yes
  ```

  This safeguard prevents disruption of existing functions that require communication among servers (such as virtual volumes).

Table 29 summarizes what happens when servers or server groups being distributed have the same names as servers or server groups on the managed server.

Table 29. Results of Configuration Refresh with Duplicate Object Names

Local definition (on managed server)	Object with duplicate name to be distributed	Result of configuration refresh
Server	Server	The local server definition is replaced by the distributed server definition only if an administrator for the managed server updated the local definition to allow replacement.
Server	Server group	The local server definition remains. The server group definition is not distributed.
Server group	Server	The local server group is deleted. The server definition is distributed.
Server group	Server group	The local server group definition is replaced by the distributed server group definition.

- Policy domains.

  When you point to a policy domain in a profile, the configuration information that will be sent to the managed servers includes the policy domain itself, and all policy sets with their associated management classes, copy groups, and client schedules in the domain. However, the ACTIVE policy set and its associated management classes, copy groups, and client schedules are not included.

  Policy domains can refer to storage pool names in the management classes, backup copy groups, and archive copy groups. As you set up the configuration information, consider whether managed servers already have or can set up or rename storage pools with these names.

  Associations between clients and schedules are not distributed with the configuration information. For clients in a managed policy domain to run client schedules, you must associate the clients with the schedules on the managed server.

  A subscribing managed server may already have a policy domain with the same name as the domain associated with the profile. The configuration refresh overwrites the domain defined on the managed server unless client nodes are already assigned to the domain. Once the domain becomes a managed object on the managed server, you can associate clients with the managed domain. Future configuration refreshes can then update the managed domain.

  If nodes are assigned to a domain with the same name as a domain being distributed, the domain is not replaced. This safeguard prevents inadvertent replacement of policy that could lead to loss of data. To replace an existing policy domain with a managed domain of the same name, you can do the following steps on the managed server:

  1. Copy the domain.
  2. Move all clients assigned to the original domain to the copied domain.
  3. Trigger a configuration refresh.
  4. Activate the appropriate policy set in the new, managed policy domain.

5. Move all clients back to the original domain, which is now managed.

- Administrative command schedules.

  When the configuration manager distributes administrative schedules, the schedules are not active on the managed server. An administrator on the managed server must activate any managed schedules to have them run on the managed server.

  A configuration refresh does not replace or remove any local schedules that are active on a managed server. However, a refresh can update an active schedule that is already managed by a configuration manager.
- Client option sets.
- TSM server scripts.

Before you can associate specific configuration information with a profile, the definitions must exist on the configuration manager. For example, to associate a policy domain named ENGDOMAIN with a profile, you must have already defined the ENGDOMAIN policy domain on the configuration manager.

Suppose you want the ALLOFFICES profile to distribute policy information from the STANDARD and ENGDOMAIN policy domains on the configuration manager. Enter the following command:

```
define profassociation alloffices domains=standard,engdomain
```

You can make the association more dynamic by specifying the special character, * (asterisk), by itself. When you specify the *, you can associate all existing objects with a profile without specifically naming them. If you later add more objects of the same type, the new objects are automatically distributed via the profile. For example, suppose you want the ADMINISTRATORS profile to distribute all administrators registered to the configuration manager. Enter the following commands on the configuration manager:

```
define profile administrators
 description='Profile to distribute administrators IDs'

define profassociation administrators admins=*
```

Whenever a managed server that is subscribed to the ADMINISTRATORS profile refreshes configuration information, it receives definitions for all administrators that exist on the configuration manager at the time of the refresh. As administrators are added, deleted, or changed on the configuration manager, the changed definitions are distributed to subscribing managed servers.

Changing a Profile

Task	Required Privilege Class
Define profile associations Update profiles	System

You can change a profile and its associated configuration information. For example, if you want to add a policy domain named FILESERVERS to objects already associated with the ALLOFFICES profile, enter the following command:

```
define profassociation alloffices domains=fileservers
```

You can also delete associated configuration information, which results in removal of configuration from the managed server. Use the DELETE PROFASSOCIATION command. See “Removing Configuration Information from Managed Servers” on page 453 for details.

On a configuration manager, you cannot directly change the names of administrators, scripts, and server groups associated with a profile. To change the name of an administrator, script, or server group associated with a profile, delete the object then define it again with a new name and associate it with the profile again. During the next configuration refresh, each managed server makes the corresponding changes in their databases.

You can change the description of the profile. Enter the following command:

```
update profile alloffices
 description='Configuration for all offices with file servers'
```

Preventing Access to Profiles While You Make Changes

If you are making changes to a profile, you may want to prevent any subscribing managed server from refreshing its configuration information until you are done. You can lock the profile to prevent access to the profile by a managed server. Locking prevents a managed server from getting information that is incomplete because you are still making changes.

Task	Required Privilege Class
Lock and unlock profiles	System

For example, to lock the ALLOFFICES profile for two hours (120 minutes), enter the following command:

```
lock profile alloffices 120
```

You can let the lock expire after two hours, or unlock the profile with the following command:

```
unlock profile alloffices
```

Distributing Changed Configuration Information

To distribute the changed profile, you can wait for each managed server to refresh its configuration to get the changed information, or you can notify each managed server from the configuration manager. Managed servers refresh profile information on a configuration refresh period. See “Refreshing Configuration Information” on page 461 for how to set this period.

Task	Required Privilege Class
Notify servers that subscribe to profiles to refresh configuration information	System

From the configuration manager, to notify all servers that are subscribers to the ALLOFFICES profile, enter the following command:

```
notify subscribers profile=alloffices
```

The managed servers then refresh their configuration information, even if the time period for refreshing the configuration has not passed.

Removing Configuration Information from Managed Servers

Task	Required Privilege Class
Delete profile associations	System

To remove configuration information from managed servers, you can do one of two things: delete the association of the object with the profile, or delete the object itself from the configuration manager.

Note: To remove all configuration information that is defined in the database of a managed server as a result of a profile subscription, you must delete the subscription using the option to discard all managed objects. See "Deleting Subscriptions" on page 460.

On the configuration manager, you can delete the association of objects with a profile. For example, you may want to remove some of the administrators that are associated with the ADMINISTRATORS profile. With an earlier command, you had included all administrators defined on the configuration manager (by specifying ADMINS=*). To change the administrators included in the profile you must first delete the association of all administrators, then associate just the administrators that you want to include. Do the following:

1. Before you make these changes, you may want to prevent any servers from refreshing their configuration until you are done. Enter the following command:
   ```
   lock profile administrators
   ```
2. Now make the change by entering the following commands:
   ```
   delete profassociation administrators admins=*

   define profassociation administrators
   admins=admin1,admin2,admin3,admin4
   ```
3. Unlock the profile:
   ```
   unlock profile administrators
   ```
4. You may want to notify any managed server that subscribes to the profile so that servers refresh their configuration information:
   ```
   notify subscribers profile=administrators
   ```

When you delete the association of an object with a profile, the configuration manager no longer distributes that object via the profile. Any managed server subscribing to the profile deletes the object from its database when it next contacts the configuration manager to refresh configuration information. However, a managed server does not delete the following objects:

- An object that is associated with another profile to which the server subscribes.
- A policy domain that has client nodes still assigned to it. To delete the domain, you must assign the affected client nodes to another policy domain on the managed server.
- An administrator that currently has a session open with the server.
- An administrator that is the last administrator with system authority on the managed server.

  Also the managed server does not change the authority of an administrator if doing so would leave the managed server without any administrators having the system privilege class.

You can avoid both problems by ensuring that you have locally defined at least one administrator with system privilege on each managed server.

- An administrative schedule that is active. To remove an active schedule, you must first make the schedule inactive on the managed server.
- A server definition for a server that currently has an open connection from the managed server.
- A server definition that is specified in the definition of a device class that is a SERVER device type.
- A server definition that is the definition for the event server for the managed server.

If you no longer need an object defined on the configuration manager itself or on any managed server, you can delete the object itself. Deleting the object itself from the configuration manager has an effect similar to deleting the association of that object with the profile: the configuration manager no longer distributes that object, and a managed server attempts to delete the object from its database when it refreshes configuration information.

Deleting Profiles

Task	Required Privilege Class
Delete profiles	System

You can delete a profile from a configuration manager. Before deleting a profile, you should ensure that no managed server still has a subscription to the profile. If the profile still has some subscribers, you should first delete the subscriptions on each managed server. When you delete subscriptions, consider whether you want the managed objects to be deleted on the managed server at the same time. For example, to delete the subscription to profile ALLOFFICES from managed server SANTIAGO without deleting the managed objects, log on to the SANTIAGO server and enter the following command:

```
delete subscription alloffices
```

Then, on the configuration manager, enter the following command:

```
delete profile alloffices
```

See “Deleting Subscriptions” on page 460 for more details about deleting subscriptions on a managed server.

Note: You can use command routing to issue the DELETE SUBSCRIPTION command for all managed servers.

If you try to delete a profile that still has subscriptions, the command fails unless you force the operation:

```
delete profile alloffices force=yes
```

If you do force the operation, managed servers that still subscribe to the deleted profile will later contact the configuration manager to try to get updates to the deleted profile. The managed servers will continue to do this until their subscriptions to the profile are deleted. A message will be issued on the managed server alerting the administrator of this condition.

Getting Information about Profiles

Task	Required Privilege Class
Request information about profiles	Any administrator

You can get information about configuration profiles defined on any configuration manager, as long as that server is defined to the server with which you are working. For example, from a configuration manager, you can display information about profiles defined on that server or on another configuration manager. From a managed server, you can display information about any profiles on the configuration manager to which the server subscribes. You can also get profile information from any other configuration manager defined to the managed server, even though the managed server does not subscribe to any of the profiles.

For example, to get information about all profiles on the HEADQUARTERS configuration manager when logged on to another server, enter the following command:

```
query profile server=headquarters
```

The following shows what the results might look like:

```
Configuration       Profile name        Locked?
manager
---------------     ---------------     -------
HEADQUARTERS        ADMINISTRATORS        No
HEADQUARTERS        DEFAULT_PROFILE       No
HEADQUARTERS        ENGINEERING           No
HEADQUARTERS        MARKETING             No
```

You may need to get detailed information about profiles and the objects associated with them, especially before subscribing to a profile. You can get the names of the objects associated with a profile by entering the following command:

```
query profile server=headquarters format=detailed
```

The following shows what the results might look like:

```
           Configuration manager: HEADQUARTERS
                    Profile name: ADMINISTRATORS
                         Locked?: No
                     Description:
           Server administrators: ADMIN1 ADMIN2 ADMIN3 ADMIN4
                  Policy domains:
Administrative command schedules: ** all objects **
          Server Command Scripts:
              Client Option Sets:
                         Servers:
                   Server Groups:

           Configuration manager: HEADQUARTERS
                    Profile name: DEFAULT_PROFILE
                         Locked?: No
                     Description:
           Server administrators:
                  Policy domains:
Administrative command schedules:
          Server Command Scripts:
              Client Option Sets:
                         Servers: ** all objects **
                   Server Groups: ** all objects **

           Configuration manager: HEADQUARTERS
                    Profile name: ENGINEERING
                         Locked?: No
                     Description:
           Server administrators:
                  Policy domains: ENGDOMAIN
Administrative command schedules:
          Server Command Scripts: QUERYALL
              Client Option Sets: DESIGNER PROGRAMMER
                         Servers:
                   Server Groups:

           Configuration manager: HEADQUARTERS
                    Profile name: MARKETING
                         Locked?: Yes
                     Description:
           Server administrators:
                  Policy domains: MARKETDOM
Administrative command schedules:
          Server Command Scripts: QUERYALL
              Client Option Sets: BASIC
                         Servers:
                   Server Groups:
```

If the server from which you issue the query is already a managed server (subscribed to one or more profiles on the configuration manager being queried), by default the query returns profile information as it is known to the managed server. Therefore the information is accurate as of the last configuration refresh done by the managed server. You may want to ensure that you see the latest version of profiles as they currently exist on the configuration manager. Enter the following command:

```
query profile uselocal=no format=detailed
```

To get more than the names of the objects associated with a profile, you can do one of the following:

- If command routing is set up between servers, you can route query commands from the server to the configuration manager. For example, to get details on the ENGDOMAIN policy domain on the HEADQUARTERS server, enter this command:

  ```
  headquarters: query domain engdomain format=detailed
  ```

You can also route commands from the configuration manager to another server to get details about definitions that already exist.

- If command routing is not set up, log on to the configuration manager and enter the query commands to get the information you need.

Subscribing to a Profile

Task	Required Privilege Class
Define subscriptions to profiles Set the period for configuration refreshes	System

After an administrator at a configuration manager has created profiles and associated objects with them, managed servers can subscribe to one or more of the profiles.

Notes:
Unless otherwise noted, the commands in this section would be run on a managed server:

1. An administrator at the managed server could issue the commands.
2. You could log in from the enterprise console and issue them.
3. If command routing is set up, you could route them from the server that you are logged in to.

After a managed server subscribes to a profile, the configuration manager sends the object definitions associated with the profile to the managed server where they are automatically stored in the database. Object definitions created this way in the database of a managed server are called managed objects. With a few exceptions, you cannot change managed objects on the managed server. The exceptions are that you can change:

- The active status of a schedule
- The lock status of an administrator
- Which policy set is active in a policy domain
- The default management class of a policy set
- The attributes of a server definition that are related to the use of virtual volumes (node name, password, and delete grace period)

Before a managed server subscribes to a profile, be aware that if you have defined any object with the same name and type as an object associated with the profile that you are subscribing to, those objects will be overwritten. You can check for such occurrences by querying the profile before subscribing to it.

When a managed server first subscribes to a profile on a configuration manager, it also automatically subscribes to DEFAULT_PROFILE, if a profile with this name is defined on the configuration manager. Unless DEFAULT_PROFILE is modified on the configuration manager, it contains all the server definitions and server groups defined on the configuration manager. In this way, all the servers in your network receive a consistent set of server and server group definitions.

Note: Although a managed server can subscribe to more than one profile on a configuration manager, it cannot subscribe to profiles on more than one configuration manager at a time.

Changes may be made to a profile, after a managed server subscribes to it. An administrator on the configuration manager can notify your server of a change by issuing the NOTIFY SUBSCRIBERS command. The configuration manager contacts each managed server having a subscription to one of the specified profiles. When a managed server is contacted, it begins refresh processing to get the configuration updates from the configuration manager.

A Subscription Scenario

This section describes a typical scenario in which a server subscribes to a profile on a configuration manager, HEADQUARTERS. In this scenario an administrator for the HEADQUARTERS server has defined three profiles, ADMINISTRATORS, ENGINEERING, and MARKETING, each with its own set of associations. In addition, DEFAULT_PROFILE was automatically defined and contains only the server and server group definitions defined on the HEADQUARTERS server. An administrator for HEADQUARTERS has given you the names of the profiles that you should be using. To subscribe to the ADMINISTRATORS and ENGINEERING profiles and keep them current, perform the following steps:

1. Display the names of the objects in the profiles on HEADQUARTERS.

   You might want to perform this step to see if the object names on the profiles are used on your server for any objects of the same type. Issue this command:

   ```
   query profile * server=headquarters format=detailed
   ```

   You might want to get detailed information on some of the objects by issuing specific query commands on either your server or the configuration manager.

   Note: If any object name matches and you subscribe to a profile containing an object with the matching name, the object on your server will be replaced, with the following exceptions:

   - A policy domain is not replaced if the domain has client nodes assigned to it.
   - An administrator with system authority is not replaced by an administrator with a lower authority level if the replacement would leave the server without a system administrator.
   - The definition of a server is not replaced unless the server definition on the managed server allows replacement.
   - A server with the same name as a server group is not replaced.
   - A locally defined, active administrative schedule is not replaced

2. Subscribe to the ADMINISTRATORS and ENGINEERING profiles.

   After the initial subscription, you do not have to specify the server name on the DEFINE SUBSCRIPTION commands. If at least one profile subscription already exists, any additional subscriptions are automatically directed to the same configuration manager. Issue these commands:

   ```
   define subscription administrators server=headquarters

   define subscription engineering
   ```

   The object definitions in these profiles are now stored on your database. In addition to ADMINISTRATORS and ENGINEERING, the server is also subscribed by default to DEFAULT_PROFILE. This means that all the server and server group definitions on HEADQUARTERS are now also stored in your database.

3. Set the time interval for obtaining refreshed configuration information from the configuration manager.

If you do not perform this step, your server checks for updates to the profiles at start up and every 60 minutes after that. Set up your server to check HEADQUARTERS for updates once a day (every 1440 minutes). If there is an update, HEADQUARTERS sends it to the managed server automatically when the server checks for updates.

```
set configrefresh 1440
```

Note: You can initiate a configuration refresh from a managed server at any time. To initiate a refresh, simply reissue the SET CONFIGREFRESH with any value greater than 0. The simplest approach is to use the current setting:

```
set configrefresh 1440
```

Querying Subscriptions

Task	Required Privilege Class
Request information about subscriptions	Any administrator
Request information about profiles	

From time to time, you may want to see what profiles a server is subscribed to. You may also want to see the last time that the configuration associated with that profile was successfully refreshed on your server. The QUERY SUBSCRIPTION command gives you this information. You can name a specific profile or use a wildcard character to display all or a subset of profiles to which the server is subscribed. For example, the following command displays ADMINISTRATORS and any other profiles that begin with the string "ADMIN":

```
query subscription admin*
```

Here is a sample of the output:

```
Configuration       Profile name           Last update
manager                                     date/time
---------------     ---------------     --------------------
HEADQUARTERS        ADMINISTRATORS      06/04/1998 17:51:49
HEADQUARTERS        ADMINS_1            06/04/1998 17:51:49
HEADQUARTERS        ADMINS_2            06/04/1998 17:51:49
```

To see what objects the ADMINISTRATORS profile contains, use the following command:

```
query profile administrators uselocal=no format=detailed
```

You will see output similar to the following:

```
           Configuration manager: HEADQUARTERS
                    Profile name: ADMINISTRATORS
                         Locked?: No
                     Description:
           Server administrators: ADMIN1 ADMIN2 ADMIN3 ADMIN4
                  Policy domains:
Administrative command schedules: ** all objects **
          Server Command Scripts:
              Client Option Sets:
                         Servers:
                   Server Groups:
```

Managed objects are stored in the database of a managed server as a result of subscriptions to profiles on a configuration manager. Any object that was created or updated in the database of the managed server as a result of a subscription has the string $$CONFIG_MANAGER$$ in place of the name of the administrator who last changed the object. For example, if the policy domain named ENGDOMAIN is a managed object and you enter this command on the managed server:

```
query domain engdomain format=detailed
```

You will see output similar to the following:

```
              Policy Domain Name: ENGDOMAIN
            Activated Policy Set:
            Activation Date/Time:
           Days Since Activation:
    Activated Default Mgmt Class:
      Number of Registered Nodes: 0
                     Description: Policy for design and software engineers
 Backup Retention (Grace Period): 30
Archive Retention (Grace Period): 365
  Last Update by (administrator): $$CONFIG_MANAGER$$
           Last Update Date/Time: 06/04/1998 17:51:49
                Managing profile: ENGINEERING
```

The field `Managing profile` shows the profile to which the managed server subscribes to get the definition of this object.

Deleting Subscriptions

Task	Required Privilege Class
Delete subscriptions to profiles	System

If you decide that a server no longer needs to subscribe to a profile, you can delete the subscription. When you delete a subscription to a profile, you can choose to discard the objects that came with the profile or keep them in your database. For example, to request that your subscription to PROFILEC be deleted and to keep the objects that came with that profile, issue the following command:

```
delete subscription profilec discardobjects=no
```

After the subscription is deleted on the managed server, the managed server issues a configuration refresh request to inform the configuration manager that the subscription is deleted. The configuration manager updates its database with the new information.

When you choose to delete objects when deleting the subscription, the server may not be able to delete some objects. For example, the server cannot delete a managed policy domain if the domain still has client nodes registered to it. The server skips objects it cannot delete, but does not delete the subscription itself. If you take no action after an unsuccessful subscription deletion, at the next configuration refresh the configuration manager will again send all the objects associated with the subscription. To successfully delete the subscription, do one of the following:

- Fix the reason that the objects were skipped. For example, reassign clients in the managed policy domain to another policy domain. After handling the skipped objects, delete the subscription again.
- Delete the subscription again, except this time do not discard the managed objects. The server can then successfully delete the subscription. However, the objects that were created because of the subscription remain.

Refreshing Configuration Information

Task	Required Privilege Class
Set the period for configuration refreshes	System (on the managed server)
Notify servers that subscribe to profiles to refresh configuration information	System (on the configuration manager)

On a configuration manager, an administrator can make changes to configuration information that is associated with a profile. How quickly the changes get distributed to a subscribing managed server depends on the configuration refresh period set on the managed server and whether the administrator on the configuration manager sent a notification.

By default, a managed server refreshes its configuration information every 60 minutes. To cause an immediate refresh, change this period. For example, to immediately refresh the configuration and change the frequency of future refreshes to once a day, enter the following command for the managed server:

```
set configrefresh 1440
```

By issuing this command with a value greater than zero, you cause the managed server to immediately start the refresh process.

At the configuration manager, you can cause managed servers to refresh their configuration information by notifying the servers. For example, to notify subscribers to all profiles, enter the following command:

```
notify subscribers profile=*
```

The managed servers then start to refresh configuration information to which they are subscribed through profiles.

A managed server automatically refreshes configuration information when it is restarted.

Handling Problems with Configuration Refresh

To monitor for any problems during a configuration refresh, you can watch the server console or activity log of the managed server. One problem that may occur is that the refresh process may skip objects. For example, a policy domain of the same name as an existing policy domain on the managed server is not distributed if the policy domain has client nodes assigned to it. See “Associating Configuration Information with a Profile” on page 449 for details on when objects cannot be distributed.

The configuration manager sends the objects that it can distribute to the managed server. The configuration manager skips (does not send) objects that conflict with local objects. If the configuration manager cannot send all objects that are associated with the profile, the managed server does not record the configuration refresh as complete. The objects that the configuration manager successfully sent are left as local instead of managed objects in the database of the managed server. The local objects left as a result of an unsuccessful configuration refresh become managed objects at the next successful configuration refresh of the same profile subscription.

Returning Managed Objects to Local Control

You may want to return one or more managed objects (objects distributed by a configuration manager via profiles) to local control on the managed servers. You can do this from the configuration manager or from the managed servers.

To do this from the configuration manager, you do not simply delete the association of the object from the profile, because that would cause the object to be deleted from subscribing managed servers. To ensure the object remains in the databases of the managed servers as a locally managed object, you can copy the current profile, make the deletion, and change the subscriptions of the managed servers to the new profile.

For example, servers are currently subscribed to the ENGINEERING profile. The ENGDOMAIN policy domain is associated with this profile. You want to return control of the ENGDOMAIN policy domain to the managed servers. You can do the following:

1. Copy the ENGINEERING profile to a new profile, ENGINEERING_B:
   ```
   copy profile engineering engineering_b
   ```
2. Delete the association of the ENGDOMAIN policy domain from ENGINEERING_B:
   ```
   delete profassociation engineering_b domains=engdomain
   ```
3. Use command routing to delete subscriptions to the ENGINEERING profile:
   ```
   americas,europe,asia: delete subscription engineering
   discardobjects=no
   ```
4. Delete the ENGINEERING profile:
   ```
   delete profile engineering
   ```
5. Use command routing to define subscriptions to the new ENGINEERING_B profile:
   ```
   americas,europe,asia: define subscription engineering_b
   ```

To return objects to local control when working on a managed server, you can delete the subscription to one or more profiles. When you delete a subscription, you can choose whether to delete the objects associated with the profile. To return objects to local control, you do not delete the objects. For example, use the following command on a managed server:

```
delete subscription engineering discardobjects=no
```

Setting Up Administrators for the Servers

Include in your profiles any administrators that you want to give access to all servers in the network. These administrators must then maintain their passwords on the configuration manager. To ensure passwords stay valid for as long as expected on all servers, set the password expiration period to the same time on all servers. One way to do this is to route a SET PASSEXP command from one server to all of the others.

Ensure that you have at least one administrator that is defined locally on each managed server with system authority. This avoids an error on configuration refresh when all administrators for a server would be removed as a result of a change to a profile on the configuration manager.

Handling Problems with Synchronization of Profiles

In rare situations, when a managed server contacts a configuration manager to refresh configuration information, the configuration manager may determine that the profile information on the two servers is not synchronized. It may appear that the configuration information is more recent on the managed server than on the configuration manager. This could occur in the following situations:

- The database on the configuration manager has been restored to an earlier time and now has configuration information from profiles that appear to be older than what the managed server has obtained.
- On the configuration manager, an administrator deleted a profile, forcing the deletion even though one or more managed servers still subscribed to the profile. The administrator redefined the profile (using the same name) before the managed server refreshed its configuration information.

If the configuration manager still has a record of the managed server's subscription to the profile, the configuration manager does not send its profile information at the next request for refreshed configuration information. The configuration manager informs the managed server that the profiles are not synchronized. The managed server then issues a message indicating this condition so that an administrator can take appropriate action. The administrator can perform the following steps:

1. If the configuration manager's database has been restored to an earlier point in time, the administrator may want to query the profile and associated objects on the managed server and then manually update the configuration manager with that information.
2. Use the DELETE SUBSCRIPTION command on the managed server to delete subscriptions to the profile that is not synchronized. If desired, you can also delete definitions of the associated objects, then define the subscription again.

It is possible that the configuration manager may not have a record of the managed server's subscription. In this case, no action is necessary. When the managed server requests a refresh of configuration information, the configuration manager sends current profile information and the managed server updates its database with that information.

Switching a Managed Server to a Different Configuration Manager

To switch a managed server from one configuration manager to another, perform the following steps:

1. Query profiles on the server that will be the new configuration manager to compare with current profiles to which the managed server subscribes.
2. On the managed server, delete all subscriptions to profiles on the current configuration manager. Remember to delete the subscription to the profile named DEFAULT_PROFILE. Consider whether to discard the managed objects in the database when you delete the subscriptions.

   Verify that all subscriptions have been deleted by querying subscriptions.
3. Change server communications as needed. Define the server that will be the new configuration manager. You can delete the server that was formerly the configuration manager.
4. On the managed server, define subscriptions to profiles on the new configuration manager.

Deleting Subscribers from a Configuration Manager

Under normal circumstances, you do not need to delete subscribers from a configuration manager. You only need to delete a subscription to a profile on the managed server (by using the DELETE SUBSCRIPTION command). When you issue the DELETE SUBSCRIPTION command, the managed server automatically notifies the configuration manager of the deletion by refreshing its configuration information. As part of the refresh process, the configuration manager is informed of the profiles that the managed server subscribes to (and does not subscribe to). If the configuration manager cannot be contacted immediately for a refresh, the configuration manager will find out that the subscription was deleted the next time the managed server refreshes configuration information.

Deleting subscribers from a configuration manager is only necessary as a way to clean up in certain unusual situations. For example, you may need to delete subscribers if a managed server goes away completely or deletes its last subscription without being able to notify the configuration manager. You then use the DELETE SUBSCRIBER command to delete all subscriptions for that subscriber (the managed server) from the configuration manager's database.

Renaming a Managed Server

To rename a managed server, perform the following steps:

1. By using command routing or by logging on to the managed server, change the name of the managed server. Use the enterprise console or use the SET SERVERNAME command.
2. Change the communication setup.
   a. On the configuration manager, delete the server definition with the old name.
   b. On the configuration manager, define the server with its new name.
3. On the managed server, refresh the configuration information. You can wait for the configuration refresh period to pass, or you can reset the refresh period to cause an immediate refresh.

Performing Tasks on Multiple Servers

To make performing tasks with multiple servers easier, TSM provides the following functions:

- Enterprise logon
- Command routing
- Server group definitions that can be used to simplify command routing

Using Tivoli Storage Manager Enterprise Logon

Enterprise logon enables the administrator's logon credentials to be used for access to other servers for successfully linking to other servers and routing commands to other servers. The administrator must be defined on each server with the appropriate administrative authority for the action or command.

Enterprise logon, in conjunction with enterprise configuration, allows an administrator to log on to one TSM server and have access to all associated TSM servers and clients that the administrator is authorized to access. Enterprise logon is available from a Web browser. The client must be configured to access a server at TSM Version 3 or later.

An administrator no longer has to remember multiple user IDs and passwords for servers and clients, other than the initial user ID and password. The administrator enters the initial user ID and password from the sign-on screen displayed on the administrator's Web browser. A single set of logon credentials are then used to verify an administrator's identity across servers and clients in a Web browser environment. Encrypted credentials ensure password security.

Authentication time-out processing requires an administrator to re-authenticate after a specific amount of time has passed. You can set the amount of time by using the SET WEBAUTHTIMEOUT command. The time-out protects against unauthorized users indefinitely accessing an unattended Web browser that has credentials stored in a Web browser cache. A pop-up is displayed on the browser that requires an administrator's ID and password to proceed.

The following can use enterprise logon:

- An administrator who uses a Web browser to connect to a TSM server
- An administrator or a help-desk person who uses a Web browser to connect to a remote client with the Web backup-archive client
- An end user of TSM who uses the Web backup-archive client to connect to their own remote client

A client can optionally disable enterprise logon.

Routing Commands

If you have set up your servers as described in "Setting Up Communications for Command Routing" on page 439, you can route TSM administrative commands to one or more servers. Command routing enables an administrator to send commands for processing to one or more servers at the same time. The output is collected and displayed at the server that issued the routed commands. A system administrator can configure and monitor many different servers from a central server by using command routing.

You can route commands to one server, multiple servers, servers defined to a named group (see "Setting Up Server Groups" on page 467), or a combination of these servers. A routed command cannot be further routed to other servers; only one level of routing is allowed.

Each server that you identify as the target of a routed command must first be defined with the DEFINE SERVER command. If a server has not been defined, that server is skipped and the command routing proceeds to the next server in the route list.

TSM does not run a routed command on the server from which you issue the command unless you also specify that server. To be able to specify the server on a routed command, you must define the server just as you did any other server.

Commands cannot be routed from the SERVER_CONSOLE ID.

Routed commands run independently on each server to which you send them. The success or failure of the command on one server does not affect the outcome on any of the other servers to which the command was sent.

For more information on command routing and return codes generated by command processing, refer to *Administrator's Reference*.

Routing Commands to One or More Servers

The following sections describe how you can route commands to one or more servers, and to server groups. To successfully route commands to other servers, you must have the proper administrative authority on all servers that receive the command for processing.

The return codes for command routing can be one of three severities: 0, ERROR, or WARNING. See *Administrator's Reference* for a list of valid return codes and severity levels.

Routing Commands to Single Servers: To route a command to a single server, enter the defined server's name, a colon, and then the command to be processed. For example, to route a QUERY STGPOOL command to the server that is named ADMIN1, enter:

```
admin1: query stgpool
```

The colon after the server name indicates the end of the routing information. This is also called the *server prefix*. Another way to indicate the server routing information is to use parentheses around the server name, as follows:

```
(admin1) query stgpool
```

Note: When writing scripts, you must use the parentheses for server routing information.

To route a command to more than one server, separate the server names with a comma. For example, to route a QUERY OCCUPANCY command to three servers named ADMIN1, GEO2, and TRADE5 enter:

```
admin1,geo2,trade5: query occupancy
```

or

```
(admin1,geo2,trade5) query occupancy
```

The command QUERY OCCUPANCY is routed to servers ADMIN1, GEO2, and TRADE5. If a server has not been defined with the DEFINE SERVER command, that server is skipped and the command routing proceeds to the next server in the route list.

The routed command output of each server is displayed in its entirety at the server that initiated command routing. In the previous example, output for ADMIN1 would be displayed, followed by the output of GEO2, and then the output of TRADE5.

Processing of a command on one server does not depend upon completion of the command processing on any other servers in the route list. For example, if GEO2 server does not successfully complete the command, the TRADE5 server continues processing the command independently.

Routing Commands to Server Groups: A server group is a named group of servers. Once you set up the groups, you can route commands to the groups. See "Setting Up Server Groups" on page 467 for how to set up a server group.

To route a QUERY STGPOOL command to the server group WEST_COMPLEX, enter:

```
west_complex: query stgpool
```

or

```
(west_complex) query stgpool
```

The QUERY STGPOOL command is sent for processing to servers BLD12 and BLD13 which are members of group WEST_COMPLEX.

To route a QUERY STGPOOL command to two server groups WEST_COMPLEX and NORTH_COMPLEX, enter:

```
west_complex,north_complex: query stgpool
```

or

```
(west_complex,north_complex) query stgpool
```

The QUERY STGPOOL command is sent for processing to servers BLD12 and BLD13 which are members of group WEST_COMPLEX, and servers NE12 and NW13 which are members of group NORTH_COMPLEX.

Routing Commands to Single Servers and Server Groups: You can route commands to multiple single servers and to server groups at the same time. For example, to route the QUERY DB command to servers HQSRV, REGSRV, and groups WEST_COMPLEX and NORTH_COMPLEX, enter:

```
hqsrv,regsrv,west_complex,north_complex: query db
```

or

```
(hqsrv,regsrv,west_complex,north_complex) query db
```

The QUERY DB command is sent for processing to servers HQSRV, REGSRV, to BLD12 and BLD13 (both members of WEST_COMPLEX), and to NE12 and NW12 (both members of NORTH_COMPLEX).

Duplicate references to servers are removed in processing. For example, if you route a command to server BLD12 and to server group WEST_COMPLEX (which includes BLD12), the command is sent only once to server BLD12.

Setting Up Server Groups

You can make command routing more efficient by creating one or more server groups and adding servers to them. You can then route commands to server groups in addition to or in place of routing commands to single servers. This section describes how to set up server groups. To use server groups, you must do the following tasks:

1. Define the server groups.
2. Add the servers as members of the appropriate group.

After you have the server groups set up, you can manage the groups and group members.

Defining a Server Group and Members of a Server Group

Task	Required Privilege Class
Define a server group Define a server group member	System

You can define groups of servers to which you can then route commands. The commands are routed to all servers in the group. To route commands to a server group you must do the following:

1. Define the server with the DEFINE SERVER command if it is not already defined (see "Setting Up Communications for Command Routing" on page 439).
2. Define a new server group with the DEFINE SERVERGROUP command. Server group names must be unique because both groups and server names are allowed for the routing information.
3. Define servers as members of a server group with the DEFINE GRPMEMBER command.

The following example shows how to create a server group named WEST_COMPLEX, and define servers BLD12 and BLD13 as members of the WEST_COMPLEX group:

```
define servergroup west_complex
define grpmember west_complex bld12,bld13
```

Managing Server Groups

You can query, copy, rename, update, and delete server groups as necessary.

Task	Required Privilege Class
Query a server group	System
Copy a server group	
Rename a server group	
Update a server group description	
Delete a server group	

Querying a Server Group: To query server group WEST_COMPLEX, enter:

```
query servergroup west_complex
```

The following is sample output from a QUERY SERVERGROUP command:

```
Server Group      Members       Description         Managing profile
---------------------------------------------------------------------------
WEST_COMPLEX      BLD12, BLD13
```

Copying a Server Group: To copy the entire server group contents of WEST_COMPLEX to a different server group named NEWWEST, enter:

```
copy servergroup west_complex newwest
```

This command creates the new group. If the new group already exists, the command fails.

Renaming a Server Group: To rename an existing server group NORTH_COMPLEX to NORTH, enter:

```
rename servergroup north_complex north
```

Updating a Server Group Description: To update the NORTH server group to modify its description, enter:

```
update servergroup north description="Northern marketing region"
```

Deleting a Server Group: To delete WEST_COMPLEX server group from the TSM server, enter:

```
delete servergroup west_complex
```

This command removes all members from the server group. The server definition for each group member is not affected. If the deleted server group is a member of other server groups, the deleted group is removed from the other groups.

Managing Group Members

You can move and delete group members from a previously defined group.

Task	Required Privilege Class
Move a group member to another group	System
Delete a group member	

Moving a Group Member to Another Group: To move group member TRADE5 from the NEWWEST group to the NORTH_COMPLEX group, enter:

```
move grpmember trade5 newwest north_complex
```

Deleting a Group Member from a Group: To delete group member BLD12 from the NEWWEST server group, enter:

```
delete grpmember newwest bld12
```

When you delete a server, the deleted server is removed from any server groups of which it was a member.

Querying Server Availability

You can test a connection from your local server to a specified server with the PING SERVER command. To ping the server GEO2, enter:

```
ping server geo2
```

The PING SERVER command uses the user ID and password of the administrative ID that issued the command. If the administrator is not defined on the server being pinged, the ping fails even if the server may be running.

Using Virtual Volumes to Store Data on Another Server

TSM lets a server (a *source server*) store the results of database backups, export operations, storage pool operations, and a DRM PREPARE command on another server (a *target server*). The data is stored as *virtual volumes*, which appear to be sequential media volumes on the source server but which are actually stored as archive files on a target server. Virtual volumes can be any of the following:

- Database backups
- Storage pool backups
- Data that is backed up, archived, or space managed from client nodes
- Client data migrated from storage pools on the source server
- Any data that can be moved by EXPORT and IMPORT commands
- DRM plan files

The source server is a client of the target server, and the data for the source server is managed only by the source server. In other words, the source server controls the expiration and deletion of the files that comprise the virtual volumes on the target server.

At the target server, the virtual volumes from the source server are seen as archive data. The source server is registered as a client node (of TYPE=SERVER) at the target server and is assigned to a policy domain. The archive copy group of the default management class of that domain specifies the storage pool for the data from the source server.

Note: If the default management class does not include an archive copy group, data cannot be stored on the target server.

Using virtual volumes can benefit you in the following ways:

- The source server can use the target server as an electronic vault for rapid recovery from a disaster.
- Smaller TSM source servers can use the storage pools and tape devices of larger TSM servers.
- For incremental database backups, it can decrease wasted space on volumes and under use of high-end tape drives.

Be aware of the following when you use virtual volumes:

- If you use virtual volumes for database backups, you might have the following situation: SERVER_A backs up its database to SERVER_B, and SERVER_B backs up its database to SERVER_A. If this is the only way databases are backed up, if both servers are at the same location, and if a disaster strikes that location, you may have no backups with which to restore your databases.
- Moving large amounts of data between the servers may slow down your communications significantly, depending on the network bandwidth and availability.
- You can specify in the device class definition (DEVTYPE=SERVER) how often and for how long a time the source server will try to contact the target server. Keep in mind that frequent attempts to contact the target server over an extended period can affect your communications.
- Under certain circumstances, inconsistencies may arise among virtual volume definitions on the source server and the archive files on the target server. You can use the RECONCILE VOLUMES command to reconcile these inconsistencies (see "Reconciling Virtual Volumes and Archive Files" on page 474 for details).
- If you want to enable data validation between a source and target server, enable the settings using both the DEFINE SERVER and REGISTER NODE commands. For more information see "Validating a Node's Data" on page 315 and *Administrator's Reference*.
- Storage space limitations on the target server affect the amount of data that you can store on that server.
- To minimize mount wait times, the total mount limit for all server definitions that specify the target server should not exceed the mount total limit at the target server. For example, a source server has two device classes, each specifying a mount limit of 2. A target server has only two tape drives. In this case, the source server mount requests could exceed the target server's tape drives.

Note: When you issue a DEFINE SERVER command, the source server sends a verification code to the target server. When the source server begins a session with the target server, it also sends the verification code. If the code matches what was previously stored on the target, the session is opened in read/write mode. If the verification code is lost at the source server (for example, after a database restore), the code can be reset by issuing an UPDATE SERVER command with the FORCESYNC=YES parameter.

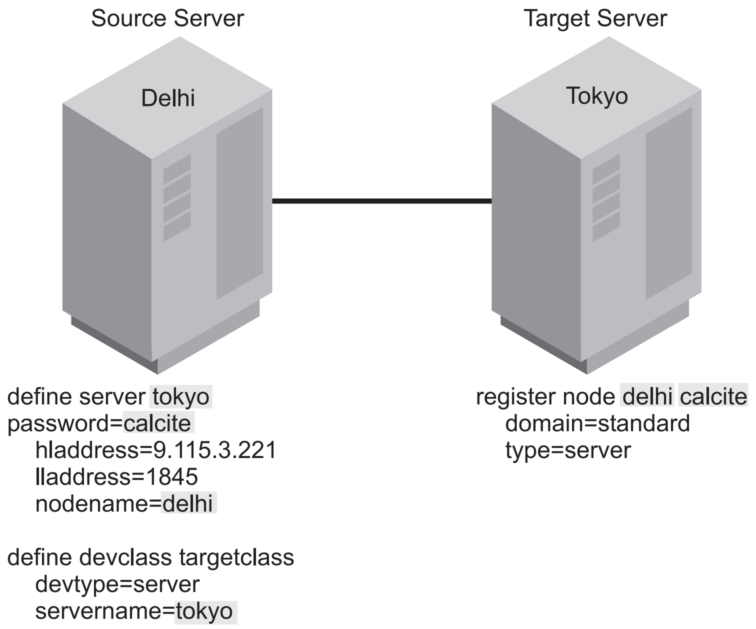
Setting Up Source and Target Servers for Virtual Volumes

In the source/target relationship, the source server is defined as a client node of the target server. To set up this relationship, a number of steps must be performed at the two servers. In the following example (illustrated in Figure 75 on page 472), the source server is named DELHI and the target server is named TOKYO.

- **At DELHI**:
  1. Define the target server:
     - TOKYO has a TCP/IP address of 9.115.3.221:1845
     - Assigns to TOKYO the password CALCITE.
     - Assigns DELHI as the node name by which the source server DELHI will be known at the target server. If no node name is assigned, the server name of the source server is used. To see the server name, you can issue the QUERY STATUS command.
  2. Define a device class for the data to be sent to the target server. The device type for this device class must be SERVER, and the definition must include the name of the target server.
- **At TOKYO:**

  Register the source server as a client node. The target server can use an existing policy domain and storage pool for the data from the source server. However, you can define a separate management policy and storage pool for the source server. Doing so can provide more control over storage pool resources.
  1. Use the REGISTER NODE command to define the source server as a node of TYPE=SERVER. The policy domain to which the node is assigned determines where the data from the source server is stored. Data from the source server is stored in the storage pool specified in the archive copy group of the default management class of that domain.
  2. You can set up a separate policy and storage pool for the source server.
     a. Define a storage pool named SOURCEPOOL:
     ```
     define stgpool sourcepool autotapeclass maxscratch=20
     ```
     b. Copy an existing policy domain STANDARD to a new domain named SOURCEDOMAIN:
     ```
     copy domain standard sourcedomain
     ```
     c. Assign SOURCEPOOL as the archive copy group destination in the default management class of SOURCEDOMAIN:
     ```
     update copygroup sourcedomain standard standard type=archive
       destination=sourcepool
     ```

     After issuing these commands, ensure that you assign the source server to the new policy domain (UPDATE NODE) and activate the policy. See "Changing Policy" on page 272 for details.



```
define server tokyo
password=calcite
   hladdress=9.115.3.221
   lladdress=1845
   nodename=delhi

define devclass targetclass
   devtype=server
   servername=tokyo
```

```
register node delhi calcite
   domain=standard
   type=server
```

Figure 75. Communication configuration for virtual volumes

Performing Operations at the Source Server

You can perform certain operations at the source server that cause data to be stored in a storage pool at the target server. These operations are:

- Database backups
- Storage pool backups
- Client data backup, archive, or migration
- Data migration from one storage pool to another
- Export of server information
- DRM prepare

The following sections describe how to perform these operations. In the examples, the following is assumed:

- The definitions shown in the previous section have been done.
- An operational TCP/IP connection exists between both servers.
- Both servers are running.

Back Up the Database

You can back up the database of a source server to a target server. For example, to perform an incremental backup of the source server and send the volumes to the target server, issue the following command:

```
backup db type=incremental devclass=targetclass
```

Expiration Processing of Database Backup Volumes and Recovery Plan Files with the Tivoli Disaster Recovery Manager: If your server is licensed for DRM, expiration processing can delete volumes containing expired database backups and recovery plan files. One or more database backup volumes may be deleted from the volume history during expiration processing if the following conditions are true:

- The volume has a device type of SERVER

- The volume is not part of the most recent database backup series
- The last volume of the database backup series has exceeded the expiration value specified with the SET DRMDBBACKUPEXPIREDAYS command

See "Moving Backup Volumes Onsite" on page 561 for more information.

You can also do an automatic database backup to a target server. For example, if you have issued the following command, a database backup occurs automatically when more than 60 percent of recovery log space is used:

```
define dbbackuptrigger devclass=targetclass logfullpct=60
```

Back Up a Storage Pool

You can back up a storage pool of a source server to a target server. For example, a primary storage pool named TAPEPOOL is on the source server. You can define a copy storage pool named TARGETCOPYPOOL, also on the source server. TARGETCOPYPOOL must have an associated device class whose device type is SERVER. When you back up TAPEPOOL to TARGETCOPYPOOL, the backup is sent to the target server. To do so, issue the following commands:

```
define stgpool targetcopypool targetclass pooltype=copy
  maxscratch=20
```

```
backup stgpool tapepool targetcopypool
```

Store Client Data on a Target Server

You can configure your TSM system so that when client nodes registered to the source server back up, archive, or migrate their data, that data is sent to the target server. When clients restore, retrieve, or recall their data, the source server gets the data from the target server.

To configure your system, ensure that the management policy for those nodes specifies a storage pool that has a device class whose device type is SERVER. For example, the following command defines the storage pool named TARGETPOOL.

```
define stgpool targetpool targetclass maxscratch=20
  reclaim=100
```

Note: Reclamation of a storage pool automatically begins when the percentage of reclaimable space, which is specified by the RECLAIM parameter, is reached. Reclamation of a target storage pool can involve the movement of a great deal of data from the target server to the source server and back to the target. If this operation occurs automatically during peak operating periods, it could slow network performance significantly. If you set the value to 100, reclamation will not occur automatically. For details about storage pool reclamation and how to begin it manually, see "Reclaiming Space in Sequential Access Storage Pools" on page 189.

Migrate Data from a Source Server Storage Pool to a Target Server Storage Pool

You can set up your storage pool hierarchy so that client data is migrated from a storage pool on the source server to the target server. For example, storage pool TAPEPOOL is on the source server. The TAPEPOOL definition specifies NEXTSTGPOOL=TARGETPOOL. TARGETPOOL has been defined on the source server as a storage pool of device type SERVER. When data is migrated from TAPEPOOL, it is sent to the target server.

```
define stgpool tapepool tapeclass nextstgpool=targetpool
  maxscratch=20
```

Export Server Information to a Target Server

You can use any of the TSM EXPORT commands to export data from one TSM source server to sequential media on a target TSM server. You must specify a device class with a device type specified as SERVER. For example, to copy server information directly to a target server, issue the following command:

```
export server devclass=targetclass
```

Import Server Information from a Target Server: If data has been exported from a source server to a target server, you can import that data from the target server to a third server. The server that will import the data uses the node ID and password of the source server to open a session with the target server. That session is in read-only mode because the third server does not have the proper verification code.

For example, to import server information from a target server, issue the following command:

```
import server devclass=targetclass
```

Reconciling Virtual Volumes and Archive Files

If you have restored the database on the source or target server, you should perform reconciliation between the virtual volumes on the source server and the archive files on the target server. You should also perform reconciliation if you have any other reason to suspect inconsistencies. For example, frequent communication errors between target and source servers could introduce a problem.

To perform reconciliation, issue the RECONCILE VOLUMES command specifying a device class of the device type of SERVER. In the following example TARGETCLASS is a server device class:

```
reconcile volumes targetclass fix=yes
```

The reconciliation action is determined by the FIX parameter as shown in the following table:

FIX=	At the Source Server	At the Target Server	Action
NO	Volumes exist	No files exist	Report error
		Files exist but are marked for deletion	
		Active files exist but attributes do not match	
	Volumes do not exist	Active files exist	Report error
		Files exist but are marked for deletion	None

FIX=	At the Source Server	At the Target Server	Action
YES	Volumes exist	No files exist	Report error **For storage pool volumes:** Mark volumes as unavailable
		Files exist but marked for deletion	Report error **For storage pool volumes:** If attributes match, mark files on the target server as active again, mark volumes on the source server as unavailable, and recommend that an AUDIT VOLUME be done to further verify the data. If attributes do not match, mark volumes as unavailable.
		Active files exist but attributes do not match	Report error **For storage pool volumes:** Mark volumes as unavailable and recommend that an AUDIT VOLUME be done to further verify the data.
	Volumes do not exist	Active files exist	Mark files for deletion on the target server.
		Files exist but marked for deletion	None

Chapter 21. Exporting and Importing Data

Tivoli Storage Manager provides an export-import facility that allows you to copy all or part of a server to removable media (export) so that data can be transferred to another server (import).

Task	Required Privilege Class
Perform export and import operations	System
Display information about export and import operations	Any administrator

This chapter takes you through the export and import tasks. See the following sections:

Concepts:
"Data That Can Be Exported and Imported"
Tasks:
"Preparing to Export or Import Data" on page 478
"Monitoring Export and Import Processes" on page 480
"Exporting Data to Sequential Media Volumes" on page 483
"Importing Data from Sequential Media Volumes" on page 486

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Data That Can Be Exported and Imported

Administrators can export or import the following types of data:

- Server control information, which includes:
  - Administrator definitions
  - Client node definitions
  - Policy and scheduling definitions
- File data from server storage, which includes file space definitions and authorization rules. You can request that file data be exported in any of the following groupings of files:
  - Active and inactive versions of backed up files, archive copies of files, and space-managed files
  - Active versions of backed up files, archive copies of files, and space-managed files
  - Active and inactive versions of backed up files
  - Active versions of backed up files
  - Archive copies of files

© Copyright IBM Corp. 1993, 2002

  - Space-managed files

Notes:

1. You can export information from an earlier version of Tivoli Storage Manager to a later one, but not from a later version to an earlier.
2. Data exported from a server with Unicode support cannot be imported to a server at an earlier version.
3. You cannot export nodes of type NAS. Export processing will exclude these nodes.

Your decision on what information to export depends on why you are exporting that information:

- To copy information to a second server (the target server), use the EXPORT NODE, EXPORT POLICY, and EXPORT ADMIN commands. This will balance the work load across servers. For example, when many client nodes access the same server, users contend for communication paths, server resources, and tape mounts during a restore or retrieve operation.

  To relieve a server of some work load and improve its performance, you may want to take one or all of the following actions:
  - Move a group of client nodes to a target server
  - Move policy definitions associated with these client nodes
  - Move administrator definitions for administrators who manage these client nodes

  When you complete the import, you can delete file spaces, client nodes, policy objects, scheduling objects and administrators from the source server. This will reduce contention for server resources.
- To copy data for the purpose of installing a new server, use the EXPORT SERVER command to copy all data to sequential media volumes.

Note: Because results could be unpredictable, ensure that expiration, migration, backup, or archive are not running when the EXPORT NODE command is issued.

Preparing to Export or Import Data

Before you export or import data, do the following:

- Use the EXPORT or IMPORT command with the PREVIEW parameter to verify what data will be moved
- Prepare sequential media for exporting and importing data

Using Preview before Exporting or Importing Data

Tivoli Storage Manager provides the PREVIEW option on the EXPORT and IMPORT commands. When PREVIEW=YES, the report shows how much data will be transferred without actually moving any data. When PREVIEW=NO, the export or import operation is performed.

Issue each EXPORT or IMPORT command with PREVIEW=YES to determine which objects and how much data will be moved. The server sends the following types of messages to the server console and to the activity log for each operation:

Export Reports the types of objects, number of objects, and number of bytes that would be copied to sequential media volumes. Use this information to determine how many sequential media volumes you will need.

Import

Reports the number and types of objects found on the sequential media volumes that meet your import specifications. Also reports information about any detected problems, such as corrupted data. Use this information to determine which data to move to the server and to determine if you have enough storage pool space allocated on the server.

To determine how much space is required to export all server data, enter:

```
export server filedata=all preview=yes
```

After you issue this command, the server starts a background process and issues a message similar to the following:

```
EXPORT SERVER started as Process 4
```

You can view the preview results on the server console or by querying the activity log.

You can request information about the background process, as described in "Requesting Information about an Export or Import Process" on page 480. If necessary, you can cancel an export or import process, as described in "Canceling Server Processes" on page 361.

Planning for Sequential Media Used to Export Data

To export data, you must specify a device class that supports sequential media and identify the volumes that will be used to store the exported data. Use this section to help you select the device classes and prepare sequential media volumes.

Selecting a Device Class

You can query the source and target servers to select a device class on each server that supports the same device type. If you cannot find a device class on each server that supports a matching device type, define a new device class for a device type that is available to both servers. See "Chapter 8. Defining Device Classes" on page 141.

Notes:

1. If the mount limit for the device class selected is reached when you request an export (that is, if all the drives are busy), the server automatically cancels lower priority operations, such as reclamation, to make a mount point available for the export.
2. You can export data to a storage pool on another server by specifying a device class whose device type is SERVER. For details, see "Using Virtual Volumes to Store Data on Another Server" on page 469.

Estimating the Number of Removable Media Volumes to Label

To estimate the number of tapes or optical disks needed to store export data, divide the number of bytes to be moved by the estimated capacity of a volume.

For example, cartridge system tape volumes used with 3490 tape devices have an estimated capacity of 360MB. If the preview shows that you need to transfer 720MB of data, label at least two tape volumes before you export the data.

Using Scratch Media

The server allows you to use scratch media to ensure that you have sufficient space to store all export data. If you use scratch media, record the label names and the order in which they were mounted. Or, use the USEDVOLUMELIST parameter on the export command to create a file containing the list of volumes used.

Labeling Removable Media Volumes

During an import process, you must specify the order in which volumes will be mounted. This order must match the order in which tapes or optical disks were mounted during the export process. To ensure that tapes or optical disks are mounted in the correct order, label tapes or optical disks with information that identifies the order in which they are mounted during the import process. For example, label tapes as DSM001, DSM002, DSM003, and so on.

When you export data, record the date and time for each labeled volume. Store this information in a safe location, because you will need the information when you import the data. Or, if you used the USEDVOLUMELIST parameter on the export command, save the resulting file. This file can be used on the import command volumes parameter.

Monitoring Export and Import Processes

The server lets you monitor export or import processes in two ways:

- You can view information about a process that is running on the server console or from an administrative client running in console mode.
- After a process has completed, you can query the activity log for status information from the server console or from an administrative client running in batch or interactive mode.

Requesting Information about an Export or Import Process

After you issue an EXPORT or IMPORT command, the server starts a background process, assigns a process ID to the operation, and displays the process ID when the operation starts.

You can query an export or import process by specifying the process ID number. For example, to request information about the EXPORT SERVER operation, which started as process 4, enter:

```
query process 4
```

If you issue a preview version of an EXPORT or IMPORT command and then query the process, the server reports the types of objects to be copied, the number of objects to be copied, and the number of bytes to be copied.

When you export or import data and then query the process, the server displays the number and types of objects copied so far, and the total number of bytes that have been transferred, along with information on any media mount requests that may be outstanding for the process.

For guidance information on querying background processes, see “Requesting Information about Server Processes” on page 403.

Viewing Information from the Server Console

When you issue an IMPORT or EXPORT command, either from the server console or from an administrative client, information is displayed on the server console. Figure 76 on page 481 shows an example of the information that is displayed after issuing an EXPORT SERVER command.

```
ANR0610I EXPORT SERVER started by SERVER_CONSOLE as process 1.
ANR0639I EXPORT SERVER: Processing domain ENGPOLDOM.
ANR0640I EXPORT SERVER: Processing policy set ACTIVE in policy domain
ENGPOLDOM.
ANR0640I EXPORT SERVER: Processing policy set STANDARD in policy domain
ENGPOLDOM.
ANR0641I EXPORT SERVER: Processing management class STANDARD in domain
ENGPOLDOM, set ACTIVE.
ANR0641I EXPORT SERVER: Processing management class STANDARD in domain
ENGPOLDOM, set STANDARD.
ANR0643I EXPORT SERVER: Processing archive copy group in domain
ENGPOLDOM, set STANDARD, management class ACTIVE.
ANR0643I EXPORT SERVER: Processing archive copy group in domain
ENGPOLDOM, set STANDARD, management class STANDARD.
ANR0643I EXPORT SERVER: Processing backup copy group in domain
ENGPOLDOM, set STANDARD, management class ACTIVE.
ANR0643I EXPORT SERVER: Processing backup copy group in domain
ENGPOLDOM, set STANDARD, management class STANDARD.
ANR0604I EXPORT SERVER: No schedules were found in policy domain * for
exporting.
ANR0635I EXPORT SERVER: Processing node TOMC.
ANR0605I EXPORT SERVER: No schedule associations were found in
policy domain * for exporting.
ANR0637I EXPORT SERVER: Processing file space DRIVED for node TOMC.
ANR0637I EXPORT SERVER: Processing file space OS2 for node TOMC.
ANR0637I EXPORT SERVER: Processing file space OS2VDISK for node TOMC.
ANR0617I EXPORT SERVER: Processing completed successfully.
ANR0620I EXPORT SERVER: Copied 1 domain(s).
ANR0621I EXPORT SERVER: Copied 2 policy set(s).
ANR0622I EXPORT SERVER: Copied 2 management class(es).
ANR0623I EXPORT SERVER: Copied 4 copy group(s).
ANR0626I EXPORT SERVER: Copied 1 node definition(s).
ANR0627I EXPORT SERVER: Copied 3 file space(s), 16 archive file(s)
and 0 backup file(s).
ANR0629I EXPORT SERVER: Copied 3045632 bytes of data.
ANR0611I EXPORT SERVER started by SERVER_CONSOLE as process 1 has ended.
```

Figure 76. Sample Export Server Output

Viewing Information from an Administrative Client

Use the console mode from an administrative client to monitor export or import operations or to capture processing messages to an output file. For example, to start an administrative session in console mode, enter:

```
> dsmadmc -consolemode
```

While the system is running in console mode, you cannot enter any administrative commands from the client session. You can, however, start another administrative client session for entering commands (for example, QUERY PROCESS) if you are using a multitasking workstation, such as AIX.

If you want the server to write all terminal output to a file, specify the OUTFILE option with a destination. For example, to write output to the SAVE.OUT file, enter:

```
> dsmadmc -consolemode -outfile=save.out
```

For information about using the CONSOLE mode option and ending an administrative session in console mode, see *Administrator's Reference*.

Querying the Activity Log for Export or Import Information

After an export or import process has completed, you can query the activity log for status information and possible error messages.

To minimize processing time when querying the activity log for export or import information, restrict the search by specifying EXPORT or IMPORT in the SEARCH parameter of the QUERY ACTLOG command.

For example, to determine how much data will be moved after issuing the preview version of the EXPORT SERVER command, query the activity log by entering:

```
query actlog search=export
```

Figure 77 displays a sample activity log report.

```
Date/Time            Message
-------------------- ---------------------------------------------------
05/03/1998 10:50:28  ANR0610I EXPORT SERVER started by ADMIN as
process 1.
05/03/1998 10:50:28  ANR0639I EXPORT SERVER: Processing domain ENGPOLDOM.
05/03/1998 10:50:28  ANR0640I EXPORT SERVER: Processing policy set
ACTIVE in policy domain ENGPOLDOM.
05/03/1998 10:50:28  ANR0640I EXPORT SERVER: Processing policy set
STANDARD in policy domain ENGPOLDOM.
05/03/1998 10:50:29  ANR0641I EXPORT SERVER: Processing management class
STANDARD in domain ENGPOLDOM, set ACTIVE.
05/03/1998 10:50:29  ANR0641I EXPORT SERVER: Processing management class
STANDARD in domain ENGPOLDOM, set STANDARD.
05/03/1998 10:50:29  ANR0643I EXPORT SERVER: Processing archive copy
group in domain ENGPOLDOM, set STANDARD, management class ACTIVE.
05/03/1998 10:50:29  ANR0643I EXPORT SERVER: Processing archive copy
group in domain ENGPOLDOM, set STANDARD, management class STANDARD.
05/03/1998 10:50:29  ANR0642I EXPORT SERVER: Processing backup copy
group in domain ENGPOLDOM, set STANDARD,  management class ACTIVE.
05/03/1998 10:50:29  ANR0642I EXPORT SERVER: Processing backup copy
group in domain ENGPOLDOM, set STANDARD, management class STANDARD.
05/03/1998 10:50:29  ANR0604I EXPORT SERVER: No schedules were found in policy
domain * for exporting.
05/03/1998 10:50:29  ANR0635I EXPORT SERVER: Processing node TOMC.
05/03/1998 10:50:29  ANR0605I EXPORT SERVER: No schedule associations were
found in policy domain * for exporting.
05/03/1998 10:50:29  ANR0637I EXPORT SERVER: Processing file space DRIVED for
node TOMC.
05/03/1998 10:50:29  ANR0637I EXPORT SERVER: Processing file space OS2 for node
TOMC.
05/03/1998 10:50:29  ANR0637I EXPORT SERVER: Processing file space OS2VDISK for
node TOMC.
05/03/1998 10:50:32  ANR0617I EXPORT SERVER: Processing completed successfully.
05/03/1998 10:50:32  ANR0620I EXPORT SERVER: Copied 1 domain(s).
05/03/1998 10:50:32  ANR0621I EXPORT SERVER: Copied 2 policy set(s).
05/03/1998 10:50:32  ANR0622I EXPORT SERVER: Copied 2 management class(es).
05/03/1998 10:50:32  ANR0623I EXPORT SERVER: Copied 4 copy group(s).
05/03/1998 10:50:32  ANR0626I EXPORT SERVER: Copied 1 node definition(s).
05/03/1998 10:50:32  ANR0627I EXPORT SERVER: Copied 3 file space(s),
16 export file(s) and 0 backup file(s).
05/03/1998 10:50:32  ANR0629I EXPORT SERVER: Copied 3045632 bytes of data.
05/03/1998 10:50:32  ANR0611I EXPORT SERVER started by ADMIN as
process 1 has ended.
```

Figure 77. Sample Activity Log Report on Exported Data

Exporting Data to Sequential Media Volumes

You can export all server control information or a subset of server control information by specifying one or more of the following export commands:

- EXPORT SERVER
- EXPORT ADMIN
- EXPORT NODE
- EXPORT POLICY

When you export data, you must specify the device class to which export data will be written. You must also list the volumes in the order in which they are to be mounted when the data is imported. See "Labeling Removable Media Volumes" on page 480 for information on labeling tape volumes.

You can specify the USEDVOLUMELIST parameter to indicate the name of a file where a list of volumes used in a successful export operation will be stored. If the specified file is created without errors, it can be used as input to the IMPORT command on the VOLUMENAMES=FILE:*filename* parameter. This file will contain comment lines with the date and time the export was done, and the command issued to create the export.

Note: An export operation will not overwrite an existing file. If you export a server or node try to export it again with the same volume name, the file is skipped and a scratch file is allocated. To use the same volume name, delete the volume entry from the volume history file.

Deciding When to Export Data

When you issue an EXPORT command, the operation runs as a background process. This process allows you to continue performing administrative tasks. In addition, users can continue to back up, archive, migrate, restore, retrieve, or recall files using the server.

If you choose to perform an export operation during normal working hours, be aware that administrators can change server definitions and users may modify files that are in server storage. If administrators or users modify data shortly after it has been exported, then the information copied to tape may not be consistent with data stored on the source server.

If you want to export an exact point-in-time copy of server control information, you can prevent administrative and other client nodes from accessing the server. See "Preventing Administrative Clients from Accessing the Server" and "Preventing Client Nodes from Accessing the Server" on page 484.

Preventing Administrative Clients from Accessing the Server

Administrators can change administrator, policy, or client node definitions during an export process. To prevent administrators from modifying these definitions, you can lock out administrator access to the server and cancel any administrative sessions before issuing an EXPORT command. After the export process is complete, unlock administrator access.

For more information on canceling sessions, see "Canceling a Tivoli Storage Manager Session" on page 257. For more information on locking or unlocking administrators from the server, see "Locking and Unlocking Administrators from the Server" on page 265.

Preventing Client Nodes from Accessing the Server

If client node information is exported while that client is backing up, archiving, or migrating files, the latest file copies for the client may not be exported to tape. To prevent users from accessing the server during export operations, cancel existing client sessions as described in “Canceling a Tivoli Storage Manager Session” on page 257. Then you can do one of the following:

- Disable server access to prevent client nodes from accessing the server, as described in “Disabling or Enabling Access to the Server” on page 259.

  This option is useful when you export all client node information from the source server and want to prevent all client nodes from accessing the server.
- Lock out particular client nodes from server access, as described in “Locking and Unlocking Client Nodes” on page 236.

  This option is useful when you export a subset of client node information from the source server and want to prevent particular client nodes from accessing the server until the export operation is complete.

After the export operation is complete, allow client nodes to access the server again by:

- Enabling the server, as described in “Disabling or Enabling Access to the Server” on page 259
- Unlocking client nodes, as described in “Locking and Unlocking Client Nodes” on page 236

Exporting Server Data

When you issue the EXPORT SERVER command, the server exports all server control information. You can also export file data information with the EXPORT SERVER command.

For example, you want to export server data to four defined tape cartridges, which are supported by the TAPECLASS device class. You want the server to use scratch volumes if the four volumes are not enough, and so you use the default of SCRATCH=YES. To issue this command, enter:

```
export server devclass=tapeclass
volumenames=dsm001,dsm002,dsm003,dsm004 filedata=all
```

During the export process, the server exports definition information before it exports file data information. This ensures that definition information is stored on the first tape volumes. This process allows you to mount a minimum number of tapes during the import process, if your goal is to copy only control information to the target server.

In the example above, the server exports:

- Administrator definitions
- Client node definitions
- Policy domain, policy set, management class, and copy group definitions
- Schedule definitions and client node associations
- File space definitions
- File space authorization rules
- Backed up, archived, and space-managed files

Exporting Administrator Information

When you issue the EXPORT ADMIN command, the server exports administrator definitions. Each administrator definition includes:

- Administrator name, password, and contact information
- Any administrative privilege classes the administrator has been granted
- Whether the administrator ID is locked from server access

You can specify a list of administrator names, or you can export all administrator names.

In the following example, definitions for the DAVEHIL and PENNER administrator IDs will be exported to the DSM001 tape volume, which is supported by the TAPECLASS device class. Do not allow any scratch media to be used during this export process. To issue this command, enter:

```
export admin davehil,penner devclass=tapeclass
volumenames=dsm001 scratch=no
```

Exporting Client Node Information

When you issue the EXPORT NODE command, the server exports client node definitions. Each client node definition includes:

- User ID, password, and contact information
- Name of the policy domain to which the client is assigned
- File compression status
- Whether the user has the authority to delete backed up or archived files from server storage
- Whether the client node ID is locked from server access

You can also specify whether to export file data. File data includes file space definitions and authorization rules. You can request that file data be exported in any of the following groupings of files:

- Active and inactive versions of backed up files, archive copies of files, and space-managed files
- Active versions of backed up files, archive copies of files, and space-managed files
- Active and inactive versions of backed up files
- Active versions of backed up files
- Archive copies of files
- Space-managed files

When client file data is exported, the server copies files to export volumes in the order of their physical location in server storage. This process minimizes the number of mounts required during the export process.

If you do not specify that you want to export file data, then the server only exports client node definitions.

For example, suppose you want to do the following:

- Export definitions for client nodes and file spaces in the ENGPOLDOM policy domain
- Export any active backup versions of files belonging to these client nodes
- Export this information to scratch volumes in the TAPECLASS device class

To issue this command, enter:

```
export node filespace=* domains=engpoldom
filedata=backupactive devclass=tapeclass
```

In this example, the server exports:

- Definitions of client nodes assigned to ENGPOLDOM
- File space definitions and backup authorizations for each client node in ENGPOLDOM
- Active versions of backed up files belonging to the client nodes assigned to ENGPOLDOM

Exporting Policy Information

When you issue the EXPORT POLICY command, the server exports the following information belonging to each specified policy domain:

- Policy domain definitions
- Policy set definitions, including the active policy set
- Management class definitions, including the default management class
- Backup copy group and archive copy group definitions
- Schedule definitions
- Associations between client nodes and schedules

For example, suppose you want to export policy and scheduling definitions from the policy domain named ENGPOLDOM. You want to use tape volumes DSM001 and DSM002, which belong to the TAPECLASS device class, but allow the server to use scratch tape volumes if necessary. To issue this command, enter:

```
export policy engpoldom
devclass=tapeclass volumenames=dsm001,dsm002
```

Importing Data from Sequential Media Volumes

Before you import data to a new target server, you must:

1. Install Tivoli Storage Manager for the target server. This step includes defining disk space for the database and recovery log.

   For information on installing Tivoli Storage Manager, see *Quick Start*.
2. Define server storage for the target server.

   Because each server operating system handles devices differently, server storage definitions are not exported. Therefore, you must define initial server storage for the target server. The target server must at least be able to use a drive that is compatible with the export media. This task can include defining libraries, drives, device classes, storage pools, and volumes. See the *Administrator's Guide* that applies to the target server.

After Tivoli Storage Manager is installed and set up on the target server, a system administrator can import all server control information or a subset of server control information by specifying one or more of the following import commands:

- IMPORT SERVER
- IMPORT ADMIN
- IMPORT NODE
- IMPORT POLICY

This section guides you through the entire process of importing all server control information and file data from tape volumes to a new target server. This process includes:

- Previewing information before you import data
- Importing definitions
- Tailoring server storage definitions on the target server
- Importing file data

After you understand how to import server control information and file data information, you can import any subset of data to the target server.

Step 1: Previewing Information before You Import Data

Before you import any data to the target server, preview each IMPORT command to determine what data you want to import to the target server. You can import all or a subset of export data from tapes.

When you set PREVIEW=YES, tape operators must mount export tape volumes so that the target server can calculate the statistics for the preview.

For example, to preview information for the IMPORT SERVER command, enter:

```
import server devclass=tapeclass preview=yes
volumenames=dsm001,dsm002,dsm003,dsm004
```

Figure 78 on page 488 shows an example of the messages sent to the server console and the activity log.

```
ANR0402I Session 3 started for administrator SERVER_CONSOLE (Server).
ANR1363I Import volume DSM001 opened (sequence number 1).
ANR0610I IMPORT SERVER started by SERVER_CONSOLE as process 2.
ANR0612I IMPORT SERVER: Reading EXPORT SERVER data from server TSM exported
05/07/1996 12:39:48.
ANR0639I IMPORT SERVER: Processing domain ENGPOLDOM.
ANR0640I IMPORT SERVER: Processing policy set ACTIVE in policy domain
ENGPOLDOM.
ANR0640I IMPORT SERVER: Processing policy set STANDARD in policy domain
ENGPOLDOM.
ANR0641I IMPORT SERVER: Processing management class STANDARD in domain
ENGPOLDOM, set ACTIVE.
ANR0641I IMPORT SERVER: Processing management class MCENG in domain
ENGPOLDOM, set STANDARD.
ANR0641I IMPORT SERVER: Processing management class STANDARD in domain
ENGPOLDOM, set STANDARD.
ANR0643I IMPORT SERVER: Processing archive copy group in domain
ENGPOLDOM, set ACTIVE, management class STANDARD.
ANR0643I IMPORT SERVER: Processing archive copy group in domain ENGPOLDOM,
set STANDARD, management class MCENG.
ANR0643I IMPORT SERVER: Processing archive copy group in domain ENGPOLDOM,
set STANDARD, management class STANDARD.
ANR0642I IMPORT SERVER: Processing backup copy group in domain ENGPOLDOM,
set ACTIVE, management class STANDARD.
ANR0642I IMPORT SERVER: Processing backup copy group in domain ENGPOLDOM,
set STANDARD, management class MCENG.
ANR0642I IMPORT SERVER: Processing backup copy group in domain ENGPOLDOM,
set STANDARD, management class STANDARD.
ANR0638I IMPORT SERVER: Processing administrator DAVEHIL.
ANR0638I IMPORT SERVER: Processing administrator PENNER.
ANR0635I IMPORT SERVER: Processing node TOMC.
ANR0636I IMPORT SERVER: Processing file space OS2 for node TOMC as file
space OS1.
ANR0636I IMPORT SERVER: Processing file space DRIVED for node TOMC as file
space DRIVE1.
ANR0636I IMPORT SERVER: Processing file space OS2VDISK for node TOMC as file
space OS2VDIS1.
ANR1365I Import volume DSM001 closed (end reached).
ANR1363I Import volume DSM002 opened (sequence number 2).
ANR1365I Import volume DSM002 closed (end reached).
ANR1363I Import volume DSM003 opened (sequence number 3).
ANR1365I Import volume DSM003 closed (end reached).
ANR1363I Import volume DSM004 opened (sequence number 4).
ANR1365I Import volume DSM004 closed (end reached).
ANR0617I IMPORT SERVER: Processing completed successfully.
ANR0620I IMPORT SERVER: Copied 1 domain(s).
ANR0621I IMPORT SERVER: Copied 2 policy set(s).
ANR0622I IMPORT SERVER: Copied 2 management class(es).
ANR0623I IMPORT SERVER: Copied 6 copy group(s).
ANR0625I IMPORT SERVER: Copied 2 administrator(s).
ANR0626I IMPORT SERVER: Copied 1 node definition(s).
ANR0627I IMPORT SERVER: Copied 3 file space(s), 0 archive file(s) and 462
backup file(s).
ANR0629I IMPORT SERVER: Copied 8856358 bytes of data.
ANR0611I IMPORT SERVER started by SERVER_CONSOLE as process 2 has ended.
```

Figure 78. Sample Report Created by Issuing Preview for an Import Server Command

Use the value reported for the total number of bytes copied to estimate storage pool space needed to store imported file data.

For example, Figure 78 shows that 8 856 358 bytes of data will be imported. Ensure that you have at least 8 856 358 bytes of available space in the backup storage pools defined to the server. You can use the QUERY STGPOOL and QUERY VOLUME commands to determine how much space is available in the server storage hierarchy.

In addition, the preview report shows that 0 archive files and 462 backup files will be imported. Because backup data is being imported, ensure that you have sufficient space in the backup storage pools used to store this backup data. See "Step 3: Tailoring Server Storage Definitions on the Target Server" on page 491 for information on identifying storage pools on the target server.

For information on specifying the PREVIEW parameter, see "Using Preview before Exporting or Importing Data" on page 478. For information on reviewing the results of a preview operation, see "Monitoring Export and Import Processes" on page 480.

Step 2: Importing Definitions

Next, you want to import server control information, which includes:

- Administrator definitions
- Client node definitions
- Policy domain, policy set, management class, and copy group definitions
- Schedule definitions and client node associations

However, do not import file data at this time, because some storage pools named in the copy group definitions may not exist yet on the target server.

Before you import server control information, do the following:

- Read and understand the following information:
  - "Determining Whether to Replace Existing Definitions"
  - "Understanding How the Server Imports Active Policy Sets"
- Start an administrative client session in console mode to capture import messages to an output file. See "Directing Import Messages to an Output File" on page 490.

Then import the server control information from specified tape volumes. See "Importing Server Control Information" on page 490.

Determining Whether to Replace Existing Definitions

By using the REPLACEDEFS parameter with the IMPORT command, you can specify whether to replace existing definitions on the target server when Tivoli Storage Manager encounters an object with the same name during the import process.

For example, if a definition exists for the ENGPOLDOM policy domain on the target server before you import policy definitions, then you must specify REPLACEDEFS=YES to replace the existing definition with the data from the export tape.

Definitions that can be replaced include administrator, client node, policy, or schedule definitions. The default is to not replace existing definitions on the target server.

Understanding How the Server Imports Active Policy Sets

When the server imports policy definitions, the following objects are imported to the target server:

- Policy domain definitions
- Policy set definitions, including the ACTIVE policy set
- Management class definitions

- Backup copy group definitions
- Archive copy group definitions
- Schedule definitions defined for each policy domain
- Client node associations, if the client node definition exists on the target server

If the server encounters a policy set named ACTIVE on the tape volume during the import process, it uses a temporary policy set named $$ACTIVE$$ to import the active policy set.

After $$ACTIVE$$ is imported to the target server, the server activates this policy set. During the activation process, the server validates the policy set by examining the management class and copy group definitions. If any of the following conditions occur, the server issues warning messages during validation:

- The storage destinations specified in the backup and archive copy groups do not refer to defined storage pools.
- The default management class does not contain a backup or archive copy group.
- The current ACTIVE policy set contains management class names that are not defined in the policy set to be activated.
- The current ACTIVE policy set contains copy group names that are not defined in the policy set to be activated.

After each $$ACTIVE$$ policy set has been activated, the server deletes that $$ACTIVE$$ policy set from the target server. To view information about active policy on the target server, you can use the following commands:

- QUERY COPYGROUP
- QUERY DOMAIN
- QUERY MGMTCLASS
- QUERY POLICYSET

Results from issuing the QUERY DOMAIN command show the activated policy set as $$ACTIVE$$. The $$ACTIVE$$ name shows you that the policy set which is currently activated for this domain is the policy set that was active at the time the export was performed.

Directing Import Messages to an Output File

The information generated by the validation process can help you define a storage hierarchy that supports the storage destinations currently defined in the import data.

You can direct import messages to an output file to capture any error messages that are detected during the import process. Do this by starting an administrative client session in console mode before you invoke the import command.

For example, to direct messages to an output file named IMPSERV.OUT, enter:

```
> dsmadmc -consolemode -outfile=impserv.out
```

Importing Server Control Information

Now you are ready to import the server control information. Based on the information generated during the preview operation, you know that all definition information has been stored on the first tape volume named DSM001. Specify that this tape volume can be read by a device belonging to the TAPECLASS device class.

From an administrative client session or from the server console, enter:

```
import server filedata=none devclass=tapeclass volumenames=dsm001
```

Step 3: Tailoring Server Storage Definitions on the Target Server

After you import definition information, use the reports generated by the import process to help you tailor storage for the target server.

To tailor server storage definitions on the target server, complete the following steps:

1. Identify any storage destinations specified in copy groups and management classes that do not match defined storage pools:
   - If the policy definitions you imported included an ACTIVE policy set, that policy set is validated and activated on the target server. Error messages generated during validation include whether any management classes or copy groups refer to storage pools that do not exist on the target server. You have a copy of these messages in a file if you directed console messages to an output file as described in "Directing Import Messages to an Output File" on page 490.
   - Query management class and copy group definitions to compare the storage destinations specified with the names of existing storage pools on the target server.

     To request detailed reports for all management classes, backup copy groups, and archive copy groups in the ACTIVE policy set, enter these commands:

     ```
     query mgmtclass * active * format=detailed
     query copygroup * active * standard type=backup format=detailed
     query copygroup * active * standard type=archive format=detailed
     ```
2. If storage destinations for management classes and copy groups in the ACTIVE policy set refer to storage pools that are not defined, do one of the following:
   - Define storage pools that match the storage destination names for the management classes and copy groups, as described in "Defining or Updating Primary Storage Pools" on page 158.
   - Change the storage destinations for the management classes and copy groups. Do the following:
     a. Copy the ACTIVE policy set to another policy set
     b. Modify the storage destinations of management classes and copy groups in that policy set, as required
     c. Activate the new policy set

     For information on copying policy sets, see "Defining and Updating a Policy Set" on page 291.

Depending on the amount of client file data that you expect to import, you may want to examine the storage hierarchy to ensure that sufficient storage space is available. Storage pools specified as storage destinations by management classes and copy groups may fill up with data. For example, you may need to define additional storage pools to which data can migrate from the initial storage destinations.

Step 4: Importing File Data Information

After you have defined the appropriate storage hierarchy on the target server, you can import file data from the tape volumes. File data includes file space definitions and authorization rules. You can request that file data be imported in any of the following groupings:

- Active and inactive versions of backed up files, archive copies of files, and space-managed files
- Active versions of backed up files, archive copies of files, and space-managed files
- Active and inactive versions of backed up files
- Active versions of backed up files
- Archive copies of files
- Space-managed files

Before you import file data information:

- Understand how the server handles duplicate file space names
- Decide whether to keep the original creation date for backup versions and archive copies or to import file data using an adjusted date

Understanding How Duplicate File Spaces Are Handled

When the server imports file data information, it imports any file spaces belonging to each specified client node. If a file space definition already exists on the target server for the node, the server does *not* replace the existing file space name.

If the server encounters duplicate file space names when it imports file data information, it creates a new file space name for the imported definition by replacing the final character or characters with a number. A message showing the old and new file space names is written to the server console and to the activity log.

For example, if the C_DRIVE and D_DRIVE file space names reside on the target server for node FRED and on the tape volume for FRED, then the server imports the C_DRIVE file space as C_DRIV1 file space and the D_DRIVE file space as D_DRIV1 file space, both assigned to node FRED.

Deciding Whether to Use a Relative Date When Importing File Data

When you import file data, you can keep the original creation date for backup versions and archive copies, or you can specify that the server use an adjusted date.

Because tape volumes containing exported data might not be used for some time, the original dates defined for backup versions and archive copies may be old enough that files are expired immediately when the data is imported to the target server.

To prevent backup versions and archive copies from being expired immediately, specify DATES=RELATIVE on the IMPORT NODE or IMPORT SERVER commands to adjust for the elapsed time since the files were exported to tape.

For example, assume that data exported to tape includes an archive copy archived five days prior to the export operation. If the tape volume resides on the shelf for six months before the data is imported to the target server, the server resets the archival date to five days prior to the import operation.

If you want to keep the original dates set for backup versions and archive copies, use DATES=ABSOLUTE, which is the default. If you use the absolute value, any files whose retention period has passed will be expired shortly after they are imported to the target server.

Issuing an Import Server or Import Node Command

You can import file data, either by issuing the IMPORT SERVER or IMPORT NODE command. When you issue either of these commands, you can specify which type of files should be imported for all client nodes specified and found on the export tapes. You can specify any of the following values to import file data:

All Specifies that all active and inactive versions of backed up files, archive copies of files, and space-managed files for specified client nodes are imported to the target server

None Specifies that no files are imported to the target server; only client node definitions are imported

Archive
Specifies that only archive copies of files are imported to the target server

Backup
Specifies that only backup copies of files, whether active or inactive, are imported to the target server

Backupactive
Specifies that only active versions of backed up files are imported to the target server

Allactive
Specifies that only active versions of backed up files, archive copies of files, and space-managed files are imported to the target server

Spacemanaged
Specifies that only files that have been migrated from a user's local file system (space-managed files) are imported

For example, suppose you want to import all backup versions of files, archive copies of files, and space-managed files to the target server. You do not want to replace any existing server control information during this import operation. Specify the four tape volumes that were identified during the preview operation. These tape volumes can be read by any device in the TAPECLASS device class. To issue this command, enter:

```
import server filedata=all replacedefs=no
devclass=tapeclass volumenames=dsm001,dsm002,dsm003,dsm004
```

You can limit the import to nodes that were assigned to specific policy domains on the source server. For example, suppose you exported from the source server the data for all nodes in all domains. To import to the target server the data only for nodes that were in the ENGDOM on the source server, enter this command:

```
import node filedata=all domains=engdom devclass=tapeclass
volumenames=dsm001,dsm002,dsm003,dsm004
```

If the ENGDOM policy domain exists on the target server, the imported nodes are assigned to that domain. If ENGDOM does not exist on the target server, the imported nodes are assigned to the STANDARD policy domain.

If you do not specify a domain on the IMPORT NODE command, the imported node is assigned to the STANDARD policy domain.

Considerations When Importing Data

You can use an import command to copy a subset of the information from export tapes to the target server. For example, if a tape was created with EXPORT SERVER, you can import only node information from the tape by using IMPORT NODE.

While the server allows you to issue any import command, data cannot be imported to the server if it has not been exported to tape. For example, if a tape is created with the EXPORT POLICY command, an IMPORT NODE command will not find any data on the tape because node information is not a subset of policy information.

Table 30 shows the commands you can use to import a subset of exported information to a target server.

Table 30. Importing a Subset of Information from Tapes

If tapes were created with this export command:	You can issue this import command:	You cannot issue this import command:
EXPORT SERVER	IMPORT SERVER IMPORT ADMIN IMPORT NODE IMPORT POLICY	—
EXPORT NODE	IMPORT NODE IMPORT SERVER	IMPORT ADMIN IMPORT POLICY
EXPORT ADMIN	IMPORT ADMIN IMPORT SERVER	IMPORT NODE IMPORT POLICY
EXPORT POLICY	IMPORT POLICY IMPORT SERVER	IMPORT ADMIN IMPORT NODE

Recovering from Errors during the Import Process

During import processing, the server may encounter invalid data due to corruption during storage on tape or in the database prior to the export operation. If invalid data is encountered during an import operation, the server does the following:

- The default value is used for the new object’s definition
- If the object already exists, the existing parameter is not changed

The server reports on the affected objects to the server console and the activity log during import and export operations. You should query these objects when the import process is complete to see if they reflect information that is acceptable.

Each time you run the IMPORT NODE or IMPORT SERVER command with the FILEDATA parameter equal to a value other than NONE, Tivoli Storage Manager creates a new file space and imports data to it. This process ensures that the current import does not overwrite data from a previous import. For information on how Tivoli Storage Manager handles duplicate file spaces, see “Understanding How Duplicate File Spaces Are Handled” on page 492.

A file space definition may already exist on the target server for the node. If so, an administrator with system privilege can issue the DELETE FILESPACE command to remove file spaces that are corrupted or no longer needed. For more information on the DELETE FILESPACE command, refer to the *Administrator’s Reference*.

Renaming a File Space

An imported file space can have the same name as a file space that already exists on a client node. In this case, the server does not overlay the existing file space, and the imported file space is given a new system generated file space name. This new name may match file space names that have not been backed up and are unknown to the server. In this case, you can use the RENAME FILESPACE command to rename the imported file space to the naming convention used for the client node.

Exporting and Importing Data from Virtual Volumes

All EXPORT and IMPORT operations described in the previous sections can also be done to virtual volumes. Data stored as virtual volumes appear to be sequential storage pool volumes on the source server, but are actually stored as archive files on another server. Those archive files can be in random or sequential access storage pools. The EXPORT and IMPORT commands are identical to those previously shown, except that the device class specified in the commands must have a device type of SERVER. For details about how to configure your server to export to or import from virtual volumes, see “Using Virtual Volumes to Store Data on Another Server” on page 469.

Part 5. Protecting the Server

© Copyright IBM Corp. 1993, 2002

Chapter 22. Protecting and Recovering Your Server

Failure or loss of the database, the recovery log, or storage pools can cause loss of client data. This chapter describes how you protect your server and, if necessary, recover your server.

Note: The term *tape* refers to any kind of sequential access, removable media unless otherwise indicated.

See the following sections:

Concepts:
“Levels of Protection” on page 500
“Storage Pool Protection: An Overview” on page 500
“Database and Recovery Log Protection: An Overview” on page 501
“Snapshot Database Protection” on page 504
“Choosing Where to Enable Data Validation” on page 532
Protecting Data:
“Mirroring the Database and Recovery Log” on page 504
“Backing Up Storage Pools” on page 507
“Using Copy Storage Pools to Improve Data Availability” on page 510
“Backing Up the Database” on page 510
“Data Validation During Audit Volume Processing” on page 531
Recovering Data:
“Recovering Your Server Using Database and Storage Pool Backups” on page 520
“Restoring Your Server Using Mirrored Volumes” on page 527
“Restoring Storage Pool Volumes” on page 527
“Auditing a Storage Pool Volume” on page 529
“Correcting Damaged Files” on page 537
Scenarios:
“Backup and Recovery Scenarios” on page 539

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Tivoli Disaster Recovery Manager (“Chapter 23. Using Tivoli Disaster Recovery Manager” on page 545) automates some tasks associated with preparing for or recovering from a disaster. This icon identifies those tasks.

© Copyright IBM Corp. 1993, 2002

Levels of Protection

For the best protection of your data, you should use all of the following:

- Backups of your storage pools
- Mirrored copies of your database and recovery log, with the recovery log mode set to roll-forward
- Full and incremental backups of your database

As an adjunct to full and incremental database backups, you can also use snapshot database backups.

Attention: ADSM Version 1 provided database salvage commands in case of a catastrophic error. Although these commands are still available, you should use the current database backup and recovery functions for the best server protection. Do not use the database salvage commands without help from an IBM service representative.

Storage Pool Protection: An Overview

If one or more storage pool volumes is lost or damaged, the client data may be permanently lost. However, you can back up storage pools to sequential access copy storage pools and move the volumes offsite. If data is lost or damaged, you can restore individual volumes or entire storage pools from the copy storage pools. The server tries to access the file from a copy storage pool if the primary copy of the file cannot be obtained for one of the following reasons:

- The primary file copy has been previously marked damaged (for information about damaged files, see “Correcting Damaged Files” on page 537).
- The primary file is stored on a volume that UNAVAILABLE or DESTROYED.
- The primary file is stored on an offline volume.
- The primary file is located in a storage pool that is UNAVAILABLE, and the operation is for restore, retrieve, or recall of files to a user, or export of file data.

For details, see “Restoring Storage Pools” on page 525, “Using Copy Storage Pools to Improve Data Availability” on page 510, “Recovering a Lost or Damaged Storage Pool Volume” on page 543, and “Maintaining the Integrity of Files” on page 537.

How Restore Processing Works

Two commands let you restore files from copy storage pools:

RESTORE STGPOOL
Restores all storage pool files that have been identified as having read errors. These files are known as *damaged* files or *unreadable* files. This command also restores all files on any volumes that have been designated as *destroyed* by using the UPDATE VOLUME command. See “Restoring Storage Pools” on page 525 for details.

RESTORE VOLUME
Recreates files that reside on a volume or volumes in the same primary storage pool. You can use this command to recreate files for one or more volumes that have been lost or damaged. See “Restoring Storage Pool Volumes” on page 527 for details.

Tivoli Storage Manager uses database information to determine which files should be restored for a volume or storage pool. As a result, restore processing does not require that the original volumes be accessed. For example, if a primary storage pool volume is damaged, you can use the RESTORE VOLUME command to recreate files that were stored on that volume, even if the volume itself is not readable. However, if you delete the damaged files (DISCARDDATA=YES on the DELETE VOLUME command), the server removes from the database references to the files on the primary storage pool volume and to copies of the files on copy storage pool volumes. You could not restore those files.

Restore processing copies files from a copy storage pool onto new primary storage pool volumes. The server then deletes database references to files on the original primary storage pool volumes. A primary storage pool volume will become empty if all files that were stored on that volume are restored to other volumes. In this case, the server automatically deletes the empty volume from the database.

How the Destroyed Volume Access Mode Works

The *destroyed* volume access mode permits the restoration of entire volumes. This mode designates primary volumes for which files are to be restored. If a volume is designated as destroyed, the server does not mount that volume for either read or write access. You can designate a volume as destroyed with either of two commands:

- The RESTORE VOLUME — This command automatically changes the access mode of specified volumes to the destroyed volume access mode using a volume list provided as part of the command.
- UPDATE VOLUME — Before using this command to restore volumes in a storage pool, you must update the access mode of the volumes to destroyed.

The destroyed designation for volumes is important during restore processing, particularly when the RESTORE STGPOOL command is used to restore a large number of primary storage pool volumes after a major disaster:

- You can designate as destroyed only those volumes that must be restored. If a volume is known to be usable after a disaster, do not set its access mode to destroyed.
- After you have identified the primary volumes to be restored and set their access mode to destroyed, you can add new volumes to the storage pool. The new volumes are used to contain the files as they are restored from the copy storage pool volumes. The new volumes can also be used for new files that end users back up, archive, or migrate.
- The designation of destroyed volumes permits tracking the files that must still be restored from copy storage pools. If restore processing is ended before completion for any reason, you can restart the restore. Only the files that still reside on destroyed volumes would need to be restored.

Database and Recovery Log Protection: An Overview

The database contains information about the client data in your storage pools. The recovery log contains records of changes to the database. If you lose the recovery log, you lose the changes that have been made since the last database backup. If you lose the database, you lose all your client data.

You have several ways to protect this information:

- Mirror the database, or the recovery log, or both.

- Back up the database to tape or remote virtual volumes. See “Using Virtual Volumes to Store Data on Another Server” on page 469.
- Back up the database to tape or remote virtual volumes. See “Using Virtual Volumes to Store Data on Another Server” on page 469 for more information. In the recovery log, save all the changes made to the database since that backup (this is called *roll-forward* mode).

Mirroring

You can prevent the loss of the database or recovery log due to a hardware failure on a single drive, by mirroring drives. Mirroring simultaneously writes the same data to multiple disks. However, mirroring does not protect against a disaster or a hardware failure that affects multiple drives or causes the loss of the entire system. While Tivoli Storage Manager is running, you can dynamically start or stop mirroring and change the capacity of the database.

Mirroring provides the following benefits:

- Protection against database and recovery log media failures
- Uninterrupted operations if a database or recovery log volume fails
- Avoidance of costly database recoveries

However, there are also costs:

- Mirroring doubles the required DASD for those volumes that are mirrored
- Mirroring results in decreased performance

Database and Recovery Log Protection

Tivoli Storage Manager can perform full and incremental database backups to tape while the server is running and available to clients. With the server running in *normal* mode, the backup media can then be stored onsite or offsite and can be used to recover the database up to the point of the backup. You can run full or incremental backups as often as needed to ensure that the database can be restored to an acceptable point-in-time.

You can provide even more complete protection if you specify *roll-forward* mode. With *roll-forward* mode and an intact recovery log, you can recover the database up to its most current state (the point at which the database was lost).

For the fastest recovery time and greatest availability of the database, mirror both the database and recovery log, and periodically back up the database. When operating in roll-forward mode, mirroring better ensures that you have an intact recovery log, which is necessary to restore the database to its most current state.

Normal Mode versus Roll-Forward Mode

Roll-forward mode offers the greatest protection for your data. However, there are costs to roll-forward mode. The following tables describe the protection afforded by each mode and the requirements for each mode.

Quality of Protection	
Normal Mode	**Roll-forward Mode**
Recover to a point-in-time of the latest full or incremental backup only.	Recover to a point-in-time of the latest full or incremental backup or, with an intact recovery log, to the most current state.
Recover the loss of client data up to the time when that data has been: • Backed up since the last database backup. • Moved due to storage pool migration, reclamation, or move data operations since the last database backup and then overwritten.	With an intact recovery log, recover to the most current state with no loss of client data.
You must restore the entire database even if only one volume is damaged.	You can restore a single volume.
	Preferable if the server supports HSM clients (space-managed files should be protected as fully as possible from hardware failure).

Storage Requirements	
Normal Mode	**Roll-forward Mode**
Does not require a recovery log to restore to a point-in-time. The recovery log keeps only uncommitted transactions, and its size is not affected by normal mode.	Requires an intact recovery log to restore to the most current state. The recovery log keeps all transactions since the last database backup. In this mode you should significantly increase the recovery log size. However: • Frequent database backups reduce recovery log storage requirements (after a backup is completed, recovery log records preceding the backup are deleted). • Mirroring the recovery log requires much less space than mirroring the database.
For the greatest availability, you should mirror the database and recovery log or place them on devices that guarantee availability.	You should mirror the recovery log to recover to the most current state. **Note:** Unlike mirroring the database, roll-forward recovery does not provide continuous operations after a media failure. This is because the database must be brought down to perform the recovery.

The following table compares four typical data recovery configurations, two for roll-forward mode and two for normal mode. In all four cases, the storage pools and the database are backed up. The benefits and costs are:

Mirroring
Whether the database and recovery log are mirrored. Mirroring costs additional disk space.

Coverage
How completely you can recover your data. Roll-forward recovery cannot be done if the recovery log is not intact. However, roll-forward mode does support point-in-time recovery.

Speed to Recover
How quickly data can be recovered.

Mode	Mirroring	Quality of Protection	Speed to Recover
Roll-Forward	Log and database	Greatest	Fastest
	Log Only	Medium	Moderate
Normal	Log and database	Medium	Moderate
	None	Least	Slowest

Attention: If the log mode is set to roll-forward after a point-in-time database restoration, a database backup starts when the server is brought up for the first time. This can cause loss of data: A tape can have current data on it, but because of the point-in-time restoration, it can be marked as scratch. When the server starts for the first time, it may use this tape to write the database backup, thus destroying the original data on this tape.

This situation could occur if roll-forward mode is enabled, but the administrator restored the database as if the server was operating in normal mode, not roll-forward mode. For example: The database is to be backed up at midnight everyday Monday through Friday. On Friday, the database was restored to a point-in-time of midnight Wednesday. Thursday's database backup was not used; this tape exists and does contain valid data. But because the database was restored to Wednesday at midnight, the Thursday's tape was marked as scratch. This tape was then inadvertently chosen and written with the database backup information. Therefore, the data for Thursday was lost.

Snapshot Database Protection

A snapshot database backup is a full database backup that does not interrupt the current full and incremental backup series. Snapshot database tapes can then be taken off-site for recovery purposes and therefore kept separate from the normal full and incremental backup tapes. For information about doing a snapshot of the database, see "Doing Snapshot Database Backups" on page 519.

Mirroring the Database and Recovery Log

Mirroring can be crucial in the recovery process. Consider the following scenario: Because of a sudden power outage, a partial page write occurs. The recovery log is corrupted and not completely readable. Without mirroring, recovery operations cannot complete when the server is restarted. However, if the recovery log is mirrored and a partial write is detected, a mirror volume can be used to construct valid images of the missing data.

This section explains how to:

- Allocate disk volumes to mirror the database and recovery log
- Define database or recovery log mirrored volume copies
- Specify mirroring and database page shadowing server options
- Request information about mirrored volumes

Task	Required Privilege Class
Define database and recovery log volumes	System or unrestricted storage
Query mirrored volumes	Any administrator

Separating Disk Volume Copies On Separate Physical Disks When Mirroring the Database and Recovery Log

By separating volume copies on different physical devices, you protect the server from media failure and increase the availability of the database and recovery log. If you cannot assign each volume copy to its own physical disk, allocate them as shown in Table 31.

Table 31. Separating Volume Copies

Physical Disk	Database Volume	Recovery Log Volume
Physical Disk 1	Database volume copy 1	Recovery log volume copy 3
Physical Disk 2	Recovery log volume copy 1	Database volume copy 2
Physical Disk 3	Database volume copy 3	Recovery log volume copy 2

Mirrored volumes must have at least the same capacity as the original volumes.

Defining Database or Recovery Log Mirrored Volume Copies

To mirror the database or recovery log, define a volume copy for each volume in the database or recovery log.

For example, the database consists of five volumes named VOL1, VOL2, VOL3, VOL4, and VOL5. To mirror the database, you must have five volumes that match the original volumes in size. Figure 79 shows a mirrored database in which VOL1–VOL5 are mirrored by VOLA–VOLE.

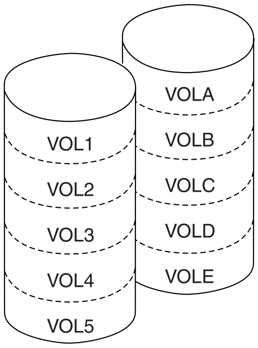


Figure 79. Mirrored Volumes

Use the DSMFMT command to format the space. For example, to format VOLA, a 25MB database volume, enter:

```
./dsmfmt -m -db vola 25
```

Then define the group of mirrored volumes. For example, you might enter the following commands:

```
define dbcopy vol1 vola

define dbcopy vol2 volb

define dbcopy vol3 volc

define dbcopy vol4 vold

define dbcopy vol5 vole
```

After a volume copy is defined, the volume copy is synchronized with the original volume. This process can range from minutes to hours, depending on the size of the volumes and performance of your system. After synchronization is complete, the volume copies are mirror images of each other.

Specifying Mirroring and Database Page Shadowing Server Options

Four server options let you specify the level of protection, recoverability, and performance for mirrored volumes:

1. MIRRORREAD specifies how mirrored volumes are accessed when the server reads the recovery log or a database page during normal processing. You may specify MIRRORREAD LOG for reading recovery log pages, or MIRRORREAD DB for reading database pages. MIRRORREAD LOG (or DB) NORMAL specifies that only one mirrored volume is read to obtain the desired page. MIRRORREAD LOG (or DB) VERIFY specifies that all mirrored volumes for a page be read, compared, and re-synchronized if necessary. MIRRORREAD LOG (or DB) VERIFY can decrease server performance as each mirrored volume for the page is accessed on every read.
2. MIRRORWRITE specifies how mirrored volumes are written to. You may issue MIRRORWRITE LOG or DB, and then specify that write operations for the database and the recovery log be specified as SEQUENTIAL or PARALLEL:
   - A PARALLEL specification offers better performance but at the potential cost of recoverability. Pages are written to all copies at about the same time. If a system outage results in a partial page write and the outage affects both mirrored copies, then both copies could be corrupted.
   - A SEQUENTIAL specification offers improved recoverability but at the cost of performance. Pages are written to one copy at a time. If a system outage results in a partial page write, only one copy is affected. However, because a successful I/O must be completed after the write to the first copy but before the write to the second copy, performance can be affected.
3. DBPAGESHADOW=YES mirrors the latest batch of pages written to a database. In this way if an outage occur that affects both mirrored volumes, the server can recover pages that have been partially written.
4. DBPAGESHADOWFILE specifies the name of the database page shadowing file.

   DBPAGESHADOW and DBPAGESHADOWFILE coordinate with the MIRRORWRITE server option and its specifications of DB and SEQUENTIAL or PARALLEL like this:

   - If database and recovery log mirroring is on with MIRRORWRITE DB PARALLEL and DBPAGESHADOW=YES, then page shadowing will be done.
   - If database and recovery log mirroring is on with MIRRORWRITE DB SEQUENTIAL, and DBPAGESHADOW=YES, then page shadowing will not be done.
   - If database and recovery log mirroring is on with MIRRORWRITE DB SEQUENTIAL and DBPAGESHADOW=NO, then page shadowing will not be done.
   - If no name is specified in the DBPAGESHADOWFILE option, a dbpgshdw.bdt file will be created and used. If the DBPAGESHADOWFILE option specifies a file name, that file name will be used.

Requesting Information about Mirrored Volumes

You can request information about mirrored database or recovery log volumes by using the QUERY DBVOLUME and QUERY LOGVOLUME commands. For example:

```
query dbvolume
```

The following type of information is displayed:

```
Volume Name    Copy    Volume Name    Copy    Volume Name    Copy
(Copy 1)       Status  (Copy 2)       Status  (Copy 3)       Status
-----------    ------  -------------  ------  -------------  ------
VOL1           Sync'd  VOLA           Sync'd                 Undef-
VOL2           Sync'd  VOLB           Sync'd                 ined
VOL3           Sync'd  VOLC           Sync'd
VOL4           Sync'd  VOLD           Sync'd

VOL5           Sync'd  VOLE           Sync'd
```

- Each pair of vertical columns displays an image of the database or recovery log. For example, VOLA, VOLB, VOLC, VOLD, and VOLE (Copy 2) represent one image of the database.
- Each horizontal row displays a *group of mirrored volumes*. For example, VOL1, and VOLA represent the two volume copies.

Backing Up Storage Pools

Task	**Required Privilege Class**
Define, back up, or restore storage pools Restore volumes	System, unrestricted storage, or restricted storage (only for those pools to which you are authorized)
Update volumes	System or operator
Query volumes or storage pools	Any administrator

You can create backup copies of client files that are stored in primary storage pools. The backup copies are stored in copy storage pools, which you can use to restore the original files if they are damaged, lost, or unusable. Primary storage pools should be backed up incrementally each day to the same copy storage pool. Backing up to the same copy storage pool ensures that files do not need to be recopied if they have migrated to the next pool.

You can back up multiple primary storage pools to one copy storage pool. If multiple copies are necessary, you can also back up a primary storage pool to multiple copy storage pools. However, you should back up the entire primary storage pool hierarchy to the same copy storage pool for easier management of storage volumes.

Note: A file that is stored in a copy storage pool may be deleted from the primary storage pool. When an incremental backup of the primary storage pool occurs, the file is deleted from the copy storage pool.

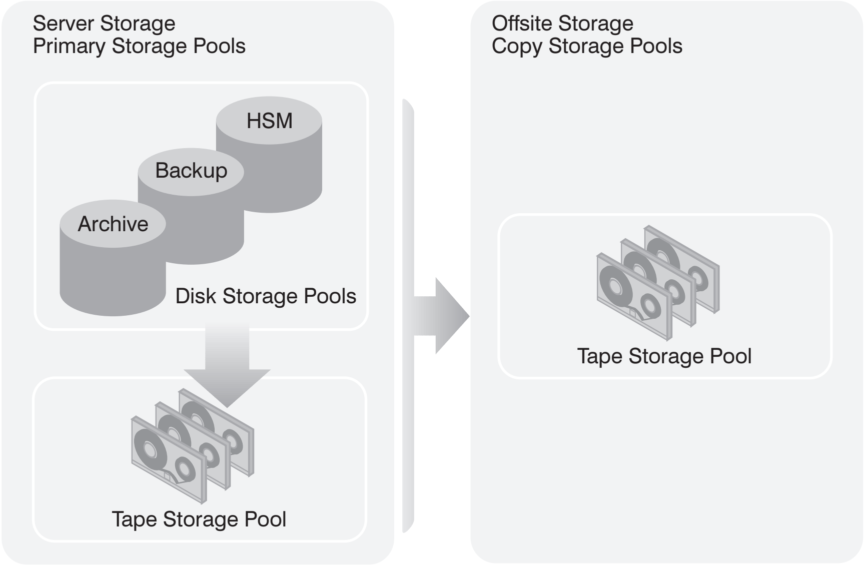


Figure 80. Copy Storage Pools

If you schedule storage pool backups and migrations and have enough disk storage, you can copy most files from the disk storage pool before they are migrated to tape and thus avoid unnecessary mounts. Here is the sequence:

1. Clients back up or archive data to disk
2. You issue the BACKUP STGPOOL command to back up the primary storage pools to copy storage pools
3. Data migrates from disk storage pools to primary tape storage pools

Backing up storage pools requires an additional 200 bytes of space in the database for each file copy. As more files are added to the copy storage pools, reevaluate your database size requirements.

Because the copies are made incrementally, you can cancel the backup process. Reissuing the BACKUP STGPOOL command lets the backup continue from the spot the backup was canceled. For example, to back up the ARCHIVEPOOL primary pool to the DISASTER-RECOVERY copy pool, enter:

```
backup stgpool archivepool disaster-recovery
```

You can define schedules to begin incremental backups of files in the primary storage pools. For example, to back up the BACKUPPOOL, ARCHIVEPOOL, and the TAPEPOOL every night, schedule the following commands:

```
backup stgpool backuppool disaster-recovery maxprocess=4

backup stgpool archivepool disaster-recovery maxprocess=4

backup stgpool tapepool disaster-recovery maxprocess=4
```

These commands use four parallel processes to perform an incremental backup of each primary storage pool to the copy pool. The only files backed up to the DISASTER-RECOVERY pool are files for which a copy does not already exist in the copy storage pool. See “Chapter 17. Automating Server Operations” on page 367 for information about scheduling commands.

Notes:

1. Set the MAXPROCESS parameter in the BACKUP STGPOOL command to the number of mount points or drives that can be dedicated to this operation.
2. Backing up storage pools requires additional space on the database.
3. If a copy is to be made in a copy storage pool and a copy already exists with the same insertion date, no action is taken.
4. Files in a copy storage pool do not migrate to another storage pool.
5. When a disk storage pool is backed up, copies of files that remain on disk after being migrated to the next storage pool (these are cached files) are not backed up.

For recovery scenarios that involve backed up copies of storage pools, see “Recovering to a Point-in-Time from a Disaster” on page 540 and “Recovering a Lost or Damaged Storage Pool Volume” on page 543.

Delaying Reuse of Sequential Access Volumes

When you define or update a sequential access storage pool, you can use the REUSEDELAY parameter. This parameter specifies the number of days that must elapse before a volume can be reused or returned to scratch status after all files have been expired, deleted, or moved from the volume. When you delay reuse of such volumes and they no longer contain any files, they enter the *pending* state. Volumes remain in the pending state for as long as specified with the REUSEDELAY parameter for the storage pool to which the volume belongs.

Delaying reuse of volumes can be helpful under certain conditions for disaster recovery. When files are expired, deleted, or moved from a volume, they are not actually erased from the volumes: The database references to these files are removed. Thus the file data may still exist on sequential volumes if the volumes are not immediately reused.

A disaster may force you to restore the database using a database backup that is not the most recent backup. In this case, some files may not be recoverable because the server cannot find them on current volumes. However, the files may exist on volumes that are in pending state. You may be able to use the volumes in pending state to recover data by doing the following:

1. Restore the database to a point-in-time prior to file expiration.
2. Use a primary or copy storage pool volume that has not been rewritten and contains the expired file at the time of database backup.

If you back up your primary storage pools, set the REUSEDELAY parameter for the primary storage pools to 0 to efficiently reuse primary scratch volumes. For your copy storage pools, you should delay reuse of volumes for as long as you keep your oldest database backup.

For an example of using database backup and delaying volume reuse, see “Protecting Your Database and Storage Pool” on page 539. For information about expiration, see “Running Expiration Processing to Delete Expired Files” on page 302.

Using Copy Storage Pools to Improve Data Availability

By using copy storage pools, you maintain multiple copies of files and reduce the potential for data loss due to media failure. If the primary file is not available or becomes corrupted, the server accesses and uses the duplicate file from a copy storage pool.

Note: To back up a primary storage pool, the DATAFORMAT must be NATIVE.

Example: Simple Hierarchy with One Copy Storage Pool

Assume that you have two primary storage pools: one random access storage pool (DISKPOOL) and one tape storage pool (TAPEPOOL, with device class TAPECLASS). Files stored in DISKPOOL are migrated to TAPEPOOL. You want to back up the files in both primary storage pools to a copy storage pool.

To schedule daily incremental backups of the primary storage pools, do the following:

1. Define a copy storage pool called COPYPOOL, with the same device class as TAPEPOOL, by issuing the following command:

   ```
   define stgpool copypool tapeclass pooltype=copy maxscratch=50
   ```

   Notes:

   a. Because scratch volumes are allowed in this copy storage pool, you do not need to define volumes for the pool.
   b. All storage volumes in COPYPOOL are located onsite.
2. Perform the initial backup of the primary storage pools by issuing the following commands:

   ```
   backup stgpool diskpool copypool
   backup stgpool tapepool copypool
   ```
3. Define schedules to automatically run the commands for backing up the primary storage pools. The commands to schedule are those that you issued in step 2.

   To minimize tape mounts, back up the disk storage pool first, then the tape storage pool.

   For more information about scheduling, see “Chapter 17. Automating Server Operations” on page 367.

Backing Up the Database

Backing up the database is a simple operation. You can back up the database with full and incremental backups or by taking a snapshot of a specific point-in-time of the database; these are called snapshot database backups. (See “Doing Full and Incremental Backups” on page 519 and “Doing Snapshot Database Backups” on page 519 for more information.) Before your first backup, you must do some or all of the following steps:

- Define device classes for backups
- Set the recovery log mode
- Schedule database backups
- Estimate the recovery log size
- Automate database backups to occur according to a defined schedule or when the recovery log utilizations reaches a specified percentage.

To restore your database, you must have copies of (or be able to create) the following information:

- Volume history file
- Device configuration file
- Server options file
- Database and recovery log set up (the output from detailed queries of your database and recovery log volumes).

DRM helps you save the previously listed information.

Defining Device Classes for Backups

You can use existing device classes for backups or define new ones. You can also specify different device classes for incremental backups and for full backups. For example, you might want to write full backups to tape and incremental backups to disk. Specifying a device class with a device type of FILE is useful if an incremental backup is run based on a database backup trigger. You should do this only if you are also backing up the files to tape and taking them off site. Otherwise, in a disaster you can only restore the full backup.

You can also reserve a device class, and therefore a device, for automatic backups only. In this way, the server does not try to back up the database with no device available. If a database backup shares a device class with a low priority operation, such as reclamation, and all the devices are in use, the lower priority operation is automatically canceled. This frees a device for the database backup.

Note: Device class definitions are saved in the device configuration files (see "Saving the Device Configuration File" on page 516).

Setting the Recovery Log Mode

You can set the recovery log mode to either *normal* or *roll-forward*. See "Database and Recovery Log Protection" on page 502 for a description of the two modes and for a comparison their benefits and costs.

If you do not set the recovery log mode, the server runs in normal mode. To set the log mode to roll-forward, enter:

```
set logmode rollforward
```

Note: The log mode is not in roll-forward mode until you perform the first full database backup after entering this command.

To set the log mode back to normal, enter:

```
set logmode normal
```

Estimating the Size of the Recovery Log

The number of transactions affect how large you should make your recovery log. As you add more clients and increase concurrent transactions, you can extend the size of the log. In roll-forward mode you should also consider how often you perform database backups. In this mode, the recovery log keeps all transactions since the last database backup and typically requires much more space than normal mode does.

To determine the size that the recovery log should be in roll-forward mode, you must know how much recovery log space is used between database backups. For example, if you perform daily incremental backups, check your daily usage over a period of time. You can use the following procedure to make your estimate:

1. Set the log mode to normal. In this way you are less likely to exceed your log space if your initial setting is too low for roll-forward mode.
2. After a scheduled database backup, reset the statistic on the amount of recovery log space used since the last reset by using the following command:
   ```
   reset logconsumption
   ```
3. Just before the next scheduled database backup, display the current recovery log statistics by using the following command:
   ```
   query log format=detailed
   ```

   Record the *cumulative consumption* value, which shows the space, in megabytes, used since the statistic was last reset.
4. Repeat steps 2 and 3 for at least one week.
5. Increase the highest cumulative consumption value by 30 percent. Set your recovery log size to this increased value to account for periods of unusually high activity.

   For example, over a period of a week the highest cumulative consumption value was 500MB. If you set your recovery log to 650MB, you should have enough space between daily backups.

For information on how to adjust the recovery log size, see “Increasing the Size of the Database or Recovery Log” on page 388 or “Decreasing the Size of the Database or Recovery Log” on page 392.

Note: If the recovery log runs out of space, you may not be able to start the server for normal operation. You can create an additional recovery log volume if needed to start the server and perform a database backup. For example, to create a 5MB volume A00, issue the following command:

```
> dsmserv extend log a00 5mb
```

Specify volume sizes in multiples of 4MB plus 1MB for overhead.

Scheduling Database Backups

Database backups require devices, media, and time. Consider scheduling backups to occur at certain times of the day and after activities such as the following:

- Major client backup or archive activities
- Storage pool migration and reclamation
- Storage pool backups
- MOVE DATA or DELETE VOLUME commands

Depending on the frequency of these activities and the amount of client data, you might back up your storage pools daily and then immediately back up the database.

Consider the following when you decide what kind of backups to do and when to do them:

- Full backups take longer than incremental backups
- Full backups have shorter recovery times than incremental backups (you must load only one set of volumes to restore the entire database)
- Full backups are required:
  - For the first backup
  - If there have been 32 incremental backups since the last full backup
  - After changing the log mode to roll-forward
  - After changing the database size (an extend or reduce operation)

Automating Database Backups

In roll-forward mode, you can set a database backup to occur automatically when the recovery log utilization reaches a defined percentage. The server also automatically deletes any unnecessary recovery log records. You might want to automate database backups if you have scheduled database backups. However, while the newly automated database backups are occurring, the recovery log could grow faster than expected. You should try to coordinate the recovery log size and scheduled backups. A database backup has a higher priority than most operations, and backup based on a trigger could occur during high server activity and affect your other operations. Adjust the recovery log size to avoid triggering backups at non-scheduled times.

By setting a database backup trigger you ensure that the recovery log does not run out of space before the next backup.

If the log mode is changed from normal to roll-forward, the next database backup must be a full backup. If a database backup trigger is defined when you set the log mode to roll-forward, the full backup is done automatically. The server does not start saving log records for roll-forward recovery until this full backup completes successfully.

By doing the steps In “Estimating the Size of the Recovery Log” on page 511, you determined the size of your recovery log. Your database backup trigger should be based on that procedure. For example, assume that your recovery log size is 650MB. Assume also that its utilization percentage is usually less than 500MB between database backups. You want to trigger a backup only in unusual circumstances. Therefore, set the trigger to at least 75 percent (approximately 500MB). To set the trigger to 75 percent and run 20 incremental backups to every full backup, enter:

```
define dbbackuptrigger logfullpct=75 devclass=tapeclass
 numincremental=20
```

Each incremental backup, whether automatic or by command, is added to the count of incremental backups. Each full backup, whether automatic or by command, resets the count for incremental backups to 0. If you specify a NUMINCREMENTAL value of 0, the server automatically runs only full backups.

Note: If you issue a BACKUP DB command with the TYPE=INCREMENTAL parameter, the server performs an incremental backup of the database regardless of the NUMINCREMENTAL setting. For example, you set NUMINCREMENTAL to 5, and there have been five incremental backups since the last full backup. If you then issue BACKUP DB TYPE=INCREMENTAL, an incremental backup is still done, and the incremental backup counter is set to 6. This occurs if the BACKUP DB command is issued either by an administrator or through an administrative schedule.

After you set the database backup trigger, you might find that automatic backups occur too often. Check the backup trigger percentage by entering:

```
query dbbackuptrigger
```

The following information is displayed:

```
            Full Device Class: TAPECLASS
     Incremental Device Class: TAPECLASS
          Log Full Percentage: 75
    Incrementals Between Fulls: 6
Last Update by (administrator): SERVER_CONSOLE
         Last Update Date/Time: 03/06/1996 10:49:23
```

This information shows that the trigger is set to 75 percent. If automatic backups are occurring too often, you could increase the value to 80 percent by entering:

```
update dbbackuptrigger logfullpct=80
```

If the database backup trigger automatically runs backups more often than you want and the setting is high (for example, 90 percent or higher), you should probably increase the recovery log size. If you no longer want to use the database backup trigger, enter:

```
delete dbbackuptrigger
```

After you delete the database backup trigger, the server no longer runs automatic database backups.

Note: If you delete the trigger and stay in roll-forward mode, transactions fail when the log fills. Therefore, you should change the log mode to normal. Remember, however, that normal mode does not let you perform roll-forward recovery. Increase the recovery log size if you want roll-forward recovery.

Saving the Volume History File

Volume history information is stored in the database, but during a database restore, it is not available from there. To perform a restore, therefore, the server must get the information from the volume history file. It is very important to save your volume history file so that you do not have to manually examine every volume.

The following volume information is stored in the database:

- Sequential access storage pool volumes that have been added, reused (through reclamation or move data operations), or deleted (during delete volume or reclamation operations)
- Full and incremental database backup volume information
- Export volumes for administrator, node, policy, and server data
- Snapshot database volume information
- Backup set volume information.

The server updates the volume history file as volumes are added. However, you must periodically run a delete operation to discard outdated information about volumes (see “Deleting Volume History Information” on page 515 for details).

To ensure the availability of volume history information, it is extremely important to do one of the following:

- Store at least one copy of the volume history file offsite or on a disk separate from the database
- Store a printout of the file offsite
- Store a copy of the file offsite with your database backups and device configuration file
- Store a remote copy of the file, for example, on an NFS-mounted file system.

DRM saves a copy of the volume history file in its disaster recovery plan file.

Note: You can recover the database without a volume history file. However, because you must examine every volume that may contain database backup information, this is a time-consuming and error-prone task.

The VOLUMEHISTORY server option lets you specify backup volume history files. Then, whenever the server updates volume information in the database, it also updates the same information in the backup files.

You can also back up the volume history information at any time, by entering:

```
backup volhistory
```

If you do not specify file names, the server backs up the volume history information to all files specified with the VOLUMEHISTORY server option.

In order to ensure updates are complete before the server is halted, we recommend you:

- Not halt the server for a few minutes after issuing the BACKUP VOLHISTORY command.
- Specify multiple VOLUMEHISTORY options in the server options file.
- Examine the volume history file to see if the file is updated.

Deleting Volume History Information

You should periodically delete outdated information from the volume history file. For example, if you keep backups for seven days, information older than that is not needed. When information about database backup volumes or export volumes is deleted, the volumes return to scratch status. For scratch volumes of device type FILE, the files are deleted. When information about storage pools volumes is deleted, the volumes themselves are not affected.

To display volume history information up to yesterday, enter:

```
query volhistory enddate=today-1
```

To delete information that is seven days old or older, enter:

```
delete volhistory type=all todate=today-8
```

Notes:

1. Existing volume history files are *not* automatically updated with the DELETE VOLHISTORY command.

2. Do not delete sequential volume history information until you no longer need that information. For example, do not delete dump volume information or storage volume reuse information, unless you have backed up or dumped the database at a later time than that specified for the delete operation.
3. Do not delete the volume history information for database dump, database backup, or export volumes that reside in automated libraries, unless you want to return the volumes to scratch status. When the DELETE VOLHISTORY command removes volume information for such volumes, they automatically return to scratch status. The volumes are then available for reuse by the server and the information stored on them may be overwritten.

DRM expires database backup series and deletes the volume history entries.

Saving the Device Configuration File

Make a copy of your device configuration file and save it.

The device configuration file contains information required to read backup data. This information includes the following:

- Devices class definitions
- Library definitions
- Drive definitions
- Server definitions

This information is stored in the database, but during a database restore, it is not available from there. To perform a restore, therefore, the server must get the information from the device configuration file. When device information is updated in the database, it is also updated in the device configuration file. The device information must match the devices configured on the system where the restore will be performed. You may have to edit those commands in an existing file so that they match.

To ensure the availability of the device configuration information, it is extremely important that you do one of the following:

- Store at least one backup copy of the device configuration file on a disk separate from the database
- Store your device configuration file offsite with your volume history file and database backups
- Store a printout of the information that is stored offsite
- Store a remote copy, for example, on an NFS-mounted file system

DRM saves a copy of the device configuration file in its disaster recovery plan file.

The DEVCONFIG server option lets you specify backup device configuration files (for details, see the *Administrator's Reference*). After the server is restarted, whenever the server updates device configuration information in the database, it also updates the same information in the backup files.

During a database restore operation, the server tries to open the first device configuration file in the order in which the files occur in the server options. If it cannot read that file, it searches for the next usable device configuration file. If none can be found, you must recreate the file. See "Recreating a Device Configuration File" on page 518 for details. After the database has been restored, you may have to update the device configuration.

You can also back up the device configuration information at any time, by entering:

```
backup devconfig
```

If you do not specify file names, the device configuration file is backed up to *all* files specified with the DEVCONFIG server option.

In order to ensure updates are complete before the server is halted, we recommend you:

- Not halt the server for a few minutes after issuing the BACKUP DEVCONFIG command.
- Specify multiple DEVCONFIG options in the server options file.
- Examine the device configuration file to see if the file is updated.

If you lose the device configuration file and need it to restore the database, you must recreate it manually. See "Recreating a Device Configuration File" on page 518 for details.

If you are using automated tape libraries, volume location information is also saved in the device configuration file. The file is updated whenever CHECKIN LIBVOLUME, CHECKOUT LIBVOLUME, and AUDIT LIBRARY commands are issued, and the information is saved as comments (/* ...... */). This information is used during restore or load operations to locate a volume in an automated library. If you must recreate the device configuration file, you will be unable to recreate the volume location information. Therefore, you must define your library as a manual library and manually mount the volumes during server processing. If an automated tape library is used at the recovery site, volume location information in comments (/*...*/) in the device configuration file must be modified. First, manually place the physical database backup volumes in the automated library and note the element numbers where you place them. Then manually edit the device configuration file to identify the locations of the database backup volumes so that the server can find them to restore the database.

For virtual volumes, the device configuration file stores the password (in encrypted form) for connecting to the remote server. If you regressed the server to an earlier point-in-time, this password may not match what the remote server expects. In this case, manually set the password in the device configuration file. Then ensure that the password on the remote server matches the password in the device configuration file.

Note: Set the password in clear text. After the server is operational again, you can issue a BACKUP DEVCONFIG command to store the password in encrypted form.

Updating the Device Configuration File

Whenever you define, update, or delete device information in the database, the device configuration file is automatically updated. This information includes definitions for device classes, libraries, drives, and servers.

If a disaster occurs, you may have to restore Tivoli Storage Manager by using devices other than those that are included in the device configuration file. In such a case, you will have to update the device configuration files manually with information about the new devices.

Recreating a Device Configuration File

The following commands read and execute the device configuration file:

- DSMSERV RESTORE DB
- DSMSERV LOADDB
- DSMSERV DISPLAY DBBACKUPVOLUME

Note: The DSMSERV LOADDB utility may increase the size of the database. The server packs data in pages in the order in which they are inserted. The DSMSERV DUMPDB utility does not preserve that order. Therefore, page packing is not optimized, and the database may require additional space.

If no device configuration file is found, you must recreate it before you can start the restore operation. The device configuration file must follow these conventions:

- The commands must be in this order:
  - DEFINE SERVER (if you are using virtual volumes)
  - DEFINE DEVCLASS
  - DEFINE LIBRARY
  - DEFINE DRIVE
  - DEFINE PATH

  You must provide those definitions needed to mount the volumes read by the command that you issued. If you are restoring or loading from a FILE device class, you will need only the DEFINE DEVCLASS command.
- For virtual volumes, the device configuration file stores the password (in encrypted form) for connecting to the remote server. If you regressed the server to an earlier point-in-time, this password may not match what the remote server expects. In this case, manually set the password in the device configuration file. Then ensure that the password on the remote server matches the password in the device configuration file.
- You can use command defaults.
- The file can include blank lines.
- A single line can be up to 240 characters.
- The file can include continuation characters and comments as described in the *Administrator's Reference*.

The following shows an example of a device configuration file:

```
/*  Tivoli Storage Manager Device Configuration */
define devclass tapeclass devtype=8mm library=manuallib
define library manuallib libtype=manual
define drive manuallib drive02
define path server1 drive02 srctype=server destype=drive
 library=manuallib device=/dev/mt2
```

Saving the Server Options

You should make a copy of your server options file and save it.

Saving the Database and Recovery Log Information

You should make copies of this output and save them. The database and recovery log setup information is the output from detailed queries of your database and recovery log volumes.

Doing Full and Incremental Backups

The first backup of your database must be a full backup. You can run up to 32 incremental backups between full backups.

To perform a full backup of your database to the TAPECLASS device class, enter:

```
backup db type=full devclass=tapeclass
```

In this example, the backup data is written to scratch volumes. You can also specify volumes by name. After a full backup, you can perform incremental backups, which copy only the changes to the database since the previous backup.

To do an incremental backup of the database to the TAPECLASS device class, enter:

```
backup db type=incremental devclass=tapeclass
```

Doing Snapshot Database Backups

A snapshot database backup is a full database backup that does not interrupt the current full and incremental backup series. Snapshot database tapes can then be taken off-site for recovery purposes and therefore kept separate from the normal full and incremental backup tapes. Snapshot database backups enhance the protection of your server and its data while maintaining the full and incremental database backup series. Although snapshot database backups cannot restore a database or a database volume to its most current state, you can use them to restore a database to a specific point-in-time.

Snapshot database backups:

- Copy the complete contents of a database, just like a full database backup.
- Create a new database backup series without interrupting the existing full and incremental backup series for the database.
- Do not truncate the server recovery log when the server is running in roll-forward mode.

Use the BACKUP DB command to perform a snapshot database backup. New volume history entries are created for the snapshot database volumes.

To perform a snapshot database backup to the TAPECLASS device class, enter:

```
backup db type=dbsnapshot devclass=tapeclass
```

Note: Snapshot database backups should be used as an adjunct to full and incremental backups. When the server is in roll-forward mode, and a snapshot database backup is performed, the recovery log keeps growing. When full and incremental backups are performed with roll-forward mode enabled, the recovery log is restarted each time a full backup is performed.

Recovering Your Server Using Database and Storage Pool Backups

This section explains how to recover by using backups of the database and storage pools. Figure 81 shows the situation presented in the two scenarios in this section: an installation has lost its server, including the database and recovery log, and its onsite storage pools.

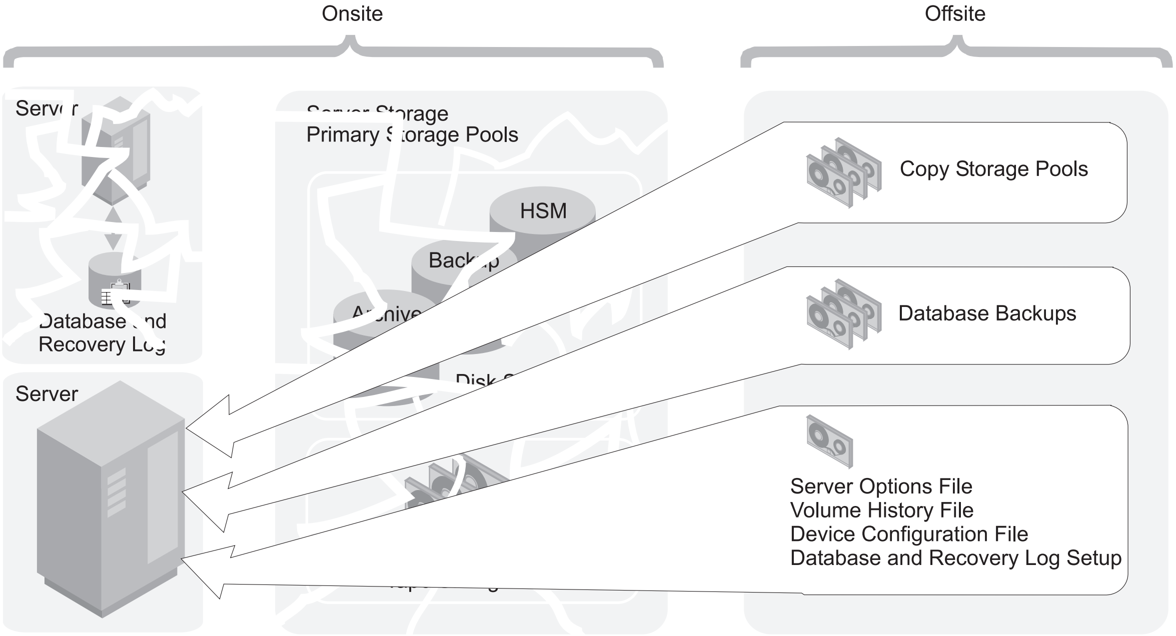


Figure 81. Recovery from a Disaster

The following topics are included:

- Restoring to a point-in-time
- Restoring to the most current state

To perform a restore, you should have the following information, preferably stored offsite (see Figure 81):

- A full database backup
- Any incremental database backups between the last full backup and the point-in-time to which you are recovering
- Copy storage pool volumes
- On tape or diskette, or as printouts:
  - Server options file
  - Volume history file
  - Device configuration file with the applicable device information (library, drive, and device class definitions)

– Database and recovery log setup (the output from detailed queries of your database and recovery log volumes)

Note: To perform a DSMSERV RESTORE DB operation, the database backup volumes must be in a library of a library type of MANUAL or SCSI.

DRM can query the server and generate a current, detailed disaster recovery plan for your installation.

Restoring a Database to a Point-in-Time

Point-in-time recovery is normally used for situations such as disaster recovery or to remove the effects of errors that can cause inconsistencies in the database.

You can use either full and incremental backups or snapshot database backups to restore a database to a point-in-time.

For a scenario of recovering to a point-in-time, see "Backup and Recovery Scenarios" on page 539.

Here is the procedure for restoring the database:

1. Rename and save a copy of the volume history file if it exists. After the database is restored, any volume history information pointed to by the server options is lost. You will need this information to identify the volumes to be audited. If you do not have a volume history file, see "Point-in-Time Restore Without a Volume History File" on page 523.
2. If the device configuration file is unavailable, recreate it manually (see "Recreating a Device Configuration File" on page 518). Put the existing or recreated device configuration file in the server work library. Do the same with the server options file. Have available your outputs from your detailed queries about your database and recovery log setup information.

   You may need to modify the device configuration file based on the hardware available at the recovery site. For example, the recovery site might require a different device class, library, and drive definitions. For more information, see "Updating the Device Configuration File" on page 518.
3. If the original database or recovery log volumes were lost, issue the DSMSERV FORMAT utility to initialize the database and recovery log. For example:

   ```
   dsmserv format 1 log1 9 1 dbvol1 5
   ```

   Attention: Do not start the server until *after* you restore the database (the next step). Starting the server before the restore would destroy any existing volume history files.
4. Issue the DSMSERV RESTORE DB utility. For example, to restore the database to a backup series that was created on April 19, 1999, enter:

   ```
   dsmserv restore db todate=04/19/1999
   ```

   The server does the following:

   a. Reads the volume history file to locate the last full backup that occurred on or before the specified date and time.

Note: If the volume history file is not available, you must mount tape volumes in the correct order or specify their order on the DSMSERV RESTORE DB utility.

b. Using the device configuration file, requests a mount of the first volume, which should contain the beginning of the full backup.

c. Restores the backup data from the first volume.

d. Continues to request mounts and to restore data from the backup volumes that contain the full backup and any incremental backups that occurred on or before the date specified.

From the old volume history information (generated by the QUERY VOLHISTORY command) you need a list of all the volumes that were reused (STGREUSE), added (STGNEW), and deleted (STGDELETE) since the original backup. Use this list to perform the rest of this procedure.

It may also be necessary to update the device configurations in the restored database.

5. Audit all disk volumes, all reused volumes, and any deleted volumes located by the AUDIT VOLUME command using the FIX=YES parameter.

   This process identifies files recorded in the database that can no longer be found on the volume. If a copy of the file is in a copy storage pool, the file on the audited volume is marked as damaged. Otherwise, the file is deleted from the database and is lost.
6. If the audit detects any damaged files, issue the RESTORE STGPOOL command to restore those files after you have audited the volumes in the storage pool. Include the FIX=YES parameter on the AUDIT VOLUME command to delete database entries for files not found in the copy storage pool.
7. Mark as destroyed any volumes that cannot be located, and recover those volumes from copy storage pool backups. If no backups are available, delete the volumes from the database by using the DELETE VOLUME command with the DISCARDDATA=YES parameter.
8. Redefine any storage pool volumes that were added since the database backup.

Notes:

1. Some files may be lost if they were moved since the backup (due to migration, reclamation, or move data requests) and the space occupied by those files has been reused. You can minimize this loss by using the REUSEDELAY parameter when defining or updating sequential access storage pools. This parameter delays volumes from being returned to scratch or being reused. See “Delaying Reuse of Sequential Access Volumes” on page 509 for more information on the REUSEDELAY parameter.
2. By backing up your storage pool and your database, you reduce the risk of losing data. To further minimize loss of data, you can:
   - Mark the backup volumes in the copy storage pool as OFFSITE and move them to an offsite location.

     In this way the backup volumes are preserved and are not reused or mounted until they are brought onsite. Ensure that you mark the volumes as OFFSITE before you back up the database.

     To avoid having to mark volumes as offsite or physically move volumes:
     – Specify a device class of DEVTYPE=SERVER in your database backup.
     – Back up a primary storage pool to a copy storage pool associated with a device class of DEVTYPE=SERVER.

- Back up the database immediately after you back up the storage pools.
- Turn off migration and reclamation while you back up the database.
- Do not perform any MOVE DATA operations while you back up the database.
- Use the REUSEDELAY parameter's interval to prevent your copy storage pool volumes from being reused or deleted before they might be needed.

3. If your old volume history file shows that any of the copy storage pool volumes needed to restore your storage pools have been reused (STGREUSE) or deleted (STGDELETE), you may not be able to restore all your files. You can avoid this problem by including the REUSEDELAY parameter when you define your copy storage pools.
4. After a restore, the volume inventories for Tivoli Storage Manager and for your tape management system may be inconsistent. For example, after a database backup, a new volume is added to Tivoli Storage Manager. The tape management system inventory records the volume as belonging to Tivoli Storage Manager. If the database is restored from the backup, Tivoli Storage Manager has no record of the added volume, but the tape management system does. You must synchronize these inventories. Similarly, the volume inventories for Tivoli Storage Manager and for any automated libraries may also be inconsistent. If they are, issue the AUDIT LIBRARY command to synchronize these inventories.

Point-in-Time Restore Without a Volume History File

You can use either full and incremental backups or snapshot database backups to restore a database to a point-in-time.

If you are doing a point-in-time restore and a volume history file is not available, you must enter the volume names in the DSMSERV RESTORE DB utility in the sequence in which they were written to. First, however, issue the DSMSERV DISPLAY DBBACKUPVOLUME utility to read your backup volumes and display the information needed to arrange them in order (backup series, backup operation, and volume sequence). For example:

```
dsmserv display dbbackupvolume devclass=tapeclass
 volumenames=dsm012,dsm023,dsm037,dsm038,dsm058,dsm087
```

For example, the most recent backup series consists of three operations:

0 A full backup on three volumes in the sequence dsm023, dsm037, and dsm087

1 An incremental backup on one volume, dsm012

2 An incremental backup on two volumes in the sequence dsm038 and dsm058

You would issue three commands in the following order:

```
dsmserv restore db volumenames=dsm023,dsm037,dsm087
 devclass=tapeclass commit=no
```

```
dsmserv restore db volumenames=dsm012
 devclass=tapeclass commit=no
```

```
dsmserv restore db volumenames=dsm038,dsm058
 devclass=tapeclass commit=no
```

Attention: If the original database or recovery log volumes are available, you issue only the DSMSERV RESTORE DB utility. However, if those volumes have been lost, you must first issue the DSMSERV FORMAT command to initialize the database and recovery log, then issue the DSMSERV RESTORE DB utility.

Storage Pool Backups in a Point-of-Time Restore

The following example shows the importance of storage pool backups with a point-in-time restore. In this example, the storage pool was not backed up with the BACKUP STGPOOL command.

9:30 a.m.
Client A backs up its data to Volume 1.

Noon The system administrator backs up the database.

1:30 p.m.
Client A's files on Volume 1 (disk), is migrated to tape (Volume 2).

3:00 p.m.
Client B backs up its data to Volume 1.

The server places Client B's files in the location that contained Client A's files prior to the migration.

3:30 p.m.
The server goes down.

3:40 p.m.
The system administrator reloads the noon version of the database by using the DSMSERV RESTORE DB utility.

4:40 p.m.
Volume 1 is audited. The following then occurs:

1. The server compares the information on Volume 1 and with the restored database (which matches the database at noon).
2. The audit does not find Client A's files on Volume 1 where the reloaded database indicates they should be. Therefore, the server deletes these Client A file references.
3. The database has no record that Client A's files are on Volume 2, and the files are, in effect, lost.
4. The database has no record that Client B's files are on Volume 1, and the files are, in effect, lost.

If roll-forward recovery had been used, the database would have been rolled forward to 3:30 p.m. when the server went down. In this case, neither Client A's files nor Client B's files would have been lost. If a point-in-time restore of the database had been performed and the storage pool had been backed up, Client A's files would not have been deleted by the volume audit. Those files could have been restored with a RESTORE VOLUME or RESTORE STGPOOL command. Client B's files would still have been lost, however.

Restoring a Database to its Most Current State

You can use roll-forward recovery to restore a database to its most current state if:

- The server has been in roll-forward mode continuously from the time of the last full backup to the time the database was damaged or lost.
- The last backup series created for the database is available. A backup series consists of a full backup, all applicable incremental backups, and all recovery log records for database changes since the last backup in the series was run.

You can only use full and incremental backups with roll-forward mode enabled to restore a database to its most current state. Snapshot database backups are complete database copies of a point-in-time.

To restore the database to its most current state, enter:

```
dsmserv restore db
```

Attention: If the original database or recovery log volumes are available, you issue only the DSMSERV RESTORE DB utility. However, if those volumes have been lost, you must first issue the DSMSERV FORMAT utility to initialize the database and recovery log, then issue the DSMSERV RESTORE DB utility.

Note: Roll-forward recovery does not apply if all recovery log volumes are lost. However, with the server running in roll-forward mode, you can still perform point-in-time recovery in such a case.

Restoring Storage Pools

You can recreate files in a primary storage pool by using duplicate copies in copy storage pools. The files must have been copied to the copy storage pools by using the BACKUP STGPOOL command.

Task	Required Privilege Class
Restoring storage pools	System, unrestricted storage, or restricted storage

The RESTORE STGPOOL command restores specified primary storage pools that have files with the following problems:

- The primary copy of the file has been identified as having read errors during a previous operation. Files with read errors are marked as damaged.
- The primary copy of the file resides on a volume that has an access mode of DESTROYED. For how the access mode of a volume changes to the DESTROYED access mode, see “How Restore Processing Works” on page 500.

When you restore a storage pool, be prepared to provide the following information:

Primary storage pool
Specifies the name of the primary storage pool that is being restored.

Copy storage pool
Specifies the name of the copy storage pool from which the files are to be restored. This information is optional. If you do not specify a copy storage pool, the server restores the files from any copy storage pool where it can find them.

New storage pool
Specifies the name of the new primary storage pool to which to restore the files. This information is optional. If you do not specify a new storage pool, the server restores the files to the original primary storage pool.

Maximum number of processes
Specifies the maximum number of parallel processes that are used for restoring files.

Preview

Specifies whether you want to preview the restore operation without actually restoring data.

See "Correcting Damaged Files" on page 537 and "Backup and Recovery Scenarios" on page 539 for examples of using the RESTORE STGPOOL command.

What Happens When a Storage Pool Is Restored

When you restore a storage pool, Tivoli Storage Manager determines which files are in that storage pool. Using file copies from a copy storage pool, Tivoli Storage Manager restores the files that were in the storage pool to the same or a different storage pool.

Cached Files: Cached copies of files in a disk storage pool are never restored. References to any cached files that have been identified as having read errors or cached files that reside on a *destroyed* volume will be removed from the database during restore processing.

The RESTORE STGPOOL command with the PREVIEW=YES parameter can be used to identify volumes that contain damaged primary files. During restore processing, a message is issued for every volume in the restored storage pool that contains damaged, noncached files. To identify the specific files that are damaged on these volumes, use the QUERY CONTENT command.

After the files are restored, the old references to these files in the primary storage pool are deleted from the database. This means that Tivoli Storage Manager now locates these files on the volumes to which they were restored, rather than on the volumes on which they were previously stored. If a destroyed volume becomes empty because all files have been restored to other locations, the destroyed volume is automatically deleted from the database.

The RESTORE STGPOOL command generates a background process that can be canceled with the CANCEL PROCESS command. If a RESTORE STGPOOL background process is canceled, some files may have already been restored prior to the cancellation. To display information about background processes, use the QUERY PROCESS command.

The RESTORE STGPOOL command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

Restoring Files to a Storage Pool with Collocation Enabled

When restoring to a primary storage pool that has collocation enabled, the server restores files by client node and client file space. This process preserves the collocation of client files. However, if the copy storage pool being used to restore files does not have collocation enabled, restore processing can be slow.

If you need to use a copy storage pool that is not collocated to restore files to a primary storage pool that is collocated, you can improve performance by:

1. Restoring the files first to a random access storage pool (on disk).
2. Allowing or forcing the files to migrate to the target primary storage pool.

   For the random access pool, set the target storage pool as the next storage pool. Adjust the migration threshold to control when migration occurs to the target storage pool.

When a Storage Pool Restoration Is Incomplete

The restoration of a storage pool volume may be incomplete. Use the QUERY CONTENT command to get more information on the remaining files on volumes for which restoration was incomplete. The restoration may be incomplete for one or more of the following reasons:

- Either files were never backed up, or the backup copies were marked as damaged.
- A copy storage pool was specified on the RESTORE command, but files were backed up to a different copy storage pool. If you suspect this is a problem, use the RESTORE command again without specifying a copy storage pool from which to restore files. The PREVIEW option can be used on the second RESTORE command, if you do not actually want to restore files.
- Volumes in the copy storage pool needed to perform the restore operation are offsite or unavailable. Check the activity log for messages that occurred during restore processing.
- Backup file copies in copy storage pools were moved or deleted by other processes during restore processing. To prevent this problem, do not issue the following commands for copy storage pool volumes while restore processing is in progress:
  - MOVE DATA
  - DELETE VOLUME (DISCARDDATA=YES)
  - AUDIT VOLUME (FIX=YES)
- You can prevent reclamation processing for your copy storage pools by setting the RECLAIM parameter to 100 with the UPDATE STGPOOL command.

Restoring Your Server Using Mirrored Volumes

If a mirrored volume fails due to media failure, you can restore the volume by taking the following steps:

1. View the status of the database and recovery log volumes by using QUERY DBVOLUME or QUERY LOGVOLUME commands.
2. If necessary, place the failing volume offline by using DELETE DBVOLUME or DELETE LOGVOLUME commands. The server usually does this automatically.
3. Fix the failing physical device.
4. Allocate space to be used for a new volume by using the DSMFMT utility.
5. Bring the volume online by using DEFINE DBCOPY or DEFINE LOGCOPY commands.

After a database or recovery log volume copy is defined, the server synchronizes the volume copy with its associated database or recovery log volume.

Restoring Storage Pool Volumes

You can recreate files in primary storage pool volumes by using copies in a copy storage pool.

Task	Required Privilege Class
Restore volumes in any storage pool for which they have authority	System, unrestricted storage, or restricted storage

Use the RESTORE VOLUME command to restore all files that are stored in the same primary storage pool and that were previously backed up to copy storage pools. When you use the RESTORE VOLUME command, be prepared to supply some or all of the following information:

Volume name
Specifies the name of the volume in the primary storage pool for which to restore files.

Tip: To restore more than one volume in the same primary storage pool, issue this command once and specify a list of volumes to be restored. When you specify more than one volume, Tivoli Storage Manager attempts to minimize volume mounts for the copy storage pool.

Copy storage pool name
Specifies the name of the copy pool from which the files are to be restored. This information is optional. If you do not specify a particular copy storage pool, the files are restored from any copy storage pool where it can find them.

New storage pool
Specifies the name of the new primary storage pool to which to restore the files. This information is optional. If you do not specify a new storage pool, the files are restored to the original primary storage pool.

Maximum number of processes
Specifies the maximum number of parallel processes that are used for restoring files.

Preview
Specifies whether you want to preview the restore operation without actually restoring data.

See “Recovering a Lost or Damaged Storage Pool Volume” on page 543 for an example of using the RESTORE VOLUME command.

What Happens When a Volume Is Restored

When you restore a volume, the server obtains a copy of each file that was on the volume from a copy storage pool, and then stores the files on a different volume.

Cached Files: Cached copies of files in a disk storage pool are never restored. References to any cached files that reside on a volume that is being restored are removed from the database during restore processing.

After files are restored, the old references to these files in the primary storage pool are deleted from the database. Tivoli Storage Manager now locates these files on the volumes to which they were restored, rather than on the volume on which they were previously stored.

The RESTORE VOLUME command changes the access mode of the volumes being restored to *destroyed*. When the restoration is complete (when all files on the volume are restored to other locations), the destroyed volume is empty and is then automatically deleted from the database.

The RESTORE VOLUME command generates a background process that can be canceled with the CANCEL PROCESS command. If a RESTORE VOLUME background process is canceled, some files may have already been restored prior to the cancellation. To display information on background processes, use the QUERY PROCESS command.

The RESTORE VOLUME command may be run in the foreground on an administrative client by issuing the command with the WAIT=YES parameter.

When a Volume Restoration Is Incomplete

The restoration of a volume may be incomplete. Use the QUERY CONTENT command to get more information on the remaining files on volumes for which restoration was incomplete. The restoration may be incomplete for one or more of the following reasons:

- Files were either never backed up or the backup copies are marked as damaged.
- A copy storage pool was specified on the RESTORE command, but files were backed up to a different copy storage pool. If you suspect this is a problem, use the RESTORE command again without specifying a copy storage pool from which to restore files. The PREVIEW option can be used on the second RESTORE command, if you do not actually want to restore files.
- Volumes in the copy storage pool needed to perform the restore operation are offsite or unavailable. Check the activity log for messages that occurred during restore processing.
- Backup file copies in copy storage pools were moved or deleted by other processes during restore processing. To prevent this problem, do not issue the following commands for copy storage pool volumes while restore processing is in progress:
  – MOVE DATA
  – DELETE VOLUME (DISCARDDATA=YES)
  – AUDIT VOLUME (FIX=YES)

  You can prevent reclamation processing for your copy storage pools by setting the RECLAIM parameter to 100 with the UPDATE STGPOOL command.

Auditing a Storage Pool Volume

Use this section to help you audit storage pool volumes for data integrity.

Task	Required Privilege Class
Audit volumes in storage pools over which they have authority	Restricted storage privilege
Audit a volume in any storage pool	System privilege, unrestricted storage privilege

The server database contains information about files on storage pool volumes. If there are inconsistencies between the information in the database about files and the files actually stored in a storage pool volume, users may be unable to access their files.

To ensure that all files are accessible on volumes in a storage pool, audit any volumes you suspect may have problems by using the AUDIT VOLUME command. You have the option of auditing multiple volumes using a time range criteria, or auditing all volumes in a storage pool.

You should audit a volume when the following conditions are true:

- The volume is damaged.
- The volume has not been accessed for a long period of time, for example, after six months
- A read or write error occurs while accessing the volume
- The database has been restored to an earlier point-in-time, and the volume is either a disk volume or a volume that was identified as being reused or deleted since the database backup

If a storage pool has data validation enabled, run an audit for the volumes in the storage pool to have the server validate the data.

What Happens When You Audit Storage Pool Volumes

When you audit a volume, a background process is started. During the auditing process, the server:

- Sends informational messages about processing to the server console.
- Prevents new files from being written to the volume.
- Generates a cyclic redundancy check, if data validation is enabled for the storage pool.
- Records results of the audit in the activity log.

You can specify whether you want the server to correct the database if inconsistencies are detected. Tivoli Storage Manager corrects the database by deleting database records that refer to files on the volume that cannot be accessed. The default is to report inconsistencies that are found (files that cannot be accessed), but to not correct the errors.

If files with read errors are detected, their handling depends on the following:

- The type of storage pool to which the volume is assigned
- The FIX option of the AUDIT VOLUME command
- The location of file copies (whether a copy of the file exists in a copy storage pool)

To display the results of a volume audit after it has completed, use the QUERY ACTLOG command. See “Requesting Information from the Activity Log” on page 412.

Volumes in a Primary Storage Pool

For a volume in a primary storage pool, the values for the FIX parameter on an AUDIT VOLUME command have the following effects:

FIX=NO

The server reports, but does not delete, any database records that refer to files found with logical inconsistencies. If the AUDIT VOLUME command detects a read error in a file, the file is marked as *damaged* in the database. You can do one of the following:

- If a backup copy of the file is stored in a copy storage pool, you can restore the file by using the RESTORE VOLUME or RESTORE STGPOOL command.
- If the file is a cached copy, you can delete references to the file on this volume by using the AUDIT VOLUME command again. Specify FIX=YES.

If the AUDIT VOLUME command does not detect a read error in a damaged file, the file state is reset, and the file can be used. For example, if a dirty tape head caused some files to be marked damaged, you can clean the head and then audit the volume to make the files accessible again.

FIX=YES

Any inconsistencies are fixed as they are detected.

If the AUDIT VOLUME command detects a read error in a file:

- If the file is not a cached copy and a backup copy is stored in a copy storage pool, the file is marked as damaged in the database. The file can then be restored using the RESTORE VOLUME or RESTORE STGPOOL command.
- If the file is not a cached copy and a backup copy is not stored in a copy storage pool, all database records that refer to the file are deleted.
- If the file is a cached copy, the database records that refer to the cached file are deleted. The primary file is stored on another volume.

If the AUDIT VOLUME command does not detect a read error in a damaged file, the file state is reset, and the file can be used. For example, if a dirty tape head caused some files to be marked damaged, you can clean the head and then audit the volume to make the files accessible again.

Volumes in a Copy Storage Pool

For volumes in a copy storage pool, the values for the FIX parameter on an AUDIT VOLUME command have the following effects:

FIX=NO

The server reports the error and marks the file copy as *damaged* in the database.

FIX=YES

The server deletes references to the file on the audited volume from the database.

Data Validation During Audit Volume Processing

Data validation for storage pools allows the server to validate that data sent to a device during a write operation matches what the server later reads. This is helpful if you have introduced new hardware devices. The validation assures that the data is not corrupt as it moves through the hardware, and then is written to the volume in the storage pool. You can use the DEFINE STGPOOL or UPDATE STGPOOL commands to enable data validation for storage pools.

When you enable data validation for an existing storage pool, the server validates data that is written from that time forward. The server does not validate existing data which was written to the storage pool before data validation was enabled.

When data validation is enabled for storage pools, the server generates a cyclic redundancy check (CRC) value and stores it with the data when it is written to the storage pool. The server validates the data when it audits the volume, by generating a cyclic redundancy check and comparing this value with the CRC value stored with the data. If the CRC values do not match, then the server processes the volume in the same manner as a standard audit volume operation. This process can depend on the following:

- The type of storage pool to which the volume is assigned
- The FIX option of the AUDIT VOLUME command

- The location of file copies (whether a copy of the file exists in a copy storage pool)

See “Volumes in a Primary Storage Pool” on page 530 and “Volumes in a Copy Storage Pool” on page 531 for details on how the server handles inconsistencies detected during an audit volume process. Check the activity log for details about the audit operation.

The server removes the CRC values before it returns the data to the client node.

Choosing Where to Enable Data Validation

Data validation is available for nodes and storage pools. The forms of validation are independent of each other. Figure 82 shows data validation:

- during a client session with the server 2
- during a client session with the storage agent 1 (the storage agent reads the VALIDATEPROTOCOL setting for the client from the Tivoli Storage Manager server)
- during a storage agent session with the server 3
- when a server (including a storage agent) sends data to the storage pool 4 or 5

You can enable data validation for one or more nodes, storage agents, or storage pools. Figure 82 illustrates data transfer that is eligible for data validation within a Tivoli Storage Manager environment. Your environment may contain some or all of these objects.

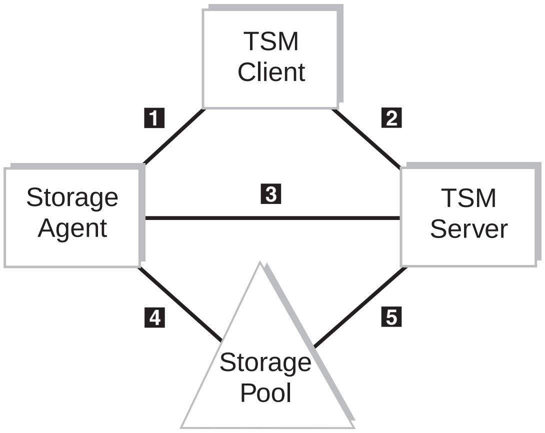


Figure 82. Data Transfer Eligible for Data Validation

Table 32 provides information that relates to Figure 82. This information explains the type of data being transferred and the appropriate command to issue.

Table 32. Setting Data Validation

Numbers in Figure 82	Where to Set Data Validation	Type of Data Transferred	Command	Command Parameter
1	Node Definition	File Data and Metadata	See ***Note***	See ***Note***

Table 32. Setting Data Validation (continued)

Numbers in Figure 82 on page 532	Where to Set Data Validation	Type of Data Transferred	Command	Command Parameter
2	Node Definition	File Data and Metadata	REGISTER NODE UPDATE NODE	VALIDATEPROTOCOL= ALL or DATAONLY
3	Server Definition (Storage Agent only)	Metadata	DEFINE SERVER UPDATE SERVER	VALIDATEPROTOCOL=ALL
4	Storage Pool Definition Issued on the TSM Server	File Data	DEFINE STGPOOL UPDATE STGPOOL	CRCDATA=YES
5	Storage Pool Definition Issued on the TSM Server	File Data	DEFINE STGPOOL UPDATE STGPOOL	CRCDATA=YES

Note: The storage agent reads the VALIDATEPROTOCOL setting for the client from the Tivoli Storage Manager server.

Figure 83 on page 534 is similar to the previous figure, however note that the top section encompassing 1, 2, and 3 is shaded. All three of these data validations are related to the VALIDATEPROTOCOL parameter. What is significant about this validation is that it is active only during the client session. After validation, the client and server discard the CRC values generated in the current session. In contrast to storage pool validation, 4 and 5, which is always active as long as the storage pool CRCDATA setting is equal to YES.

An additional aspect of storage pool validation, that you should understand, is that the validation of data transfer between the storage pool and the storage agent 4 is managed by the storage pool CRCDATA setting defined by the TSM server. So even though the flow of data is between the storage agent and the storage pool, data validation is determined by the TSM server storage pool definition. Therefore, if you always want your storage pool data validated, set your primary storage pool CRCDATA setting to YES.

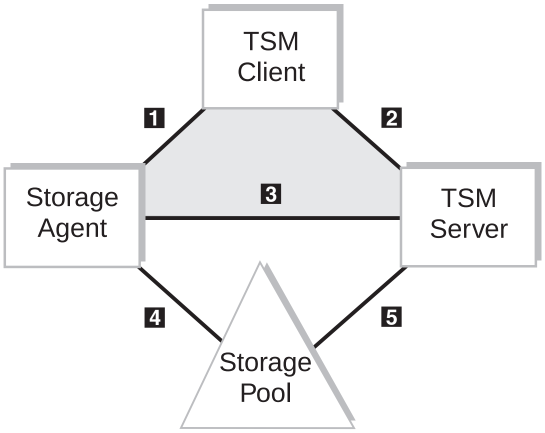


Figure 83. Protocol Data Validation versus Storage Pool Data Validation

If the network is unstable, you may decide to only enable data validation for nodes. Tivoli Storage Manager generates a cyclic redundancy check when the data is sent over the network to the server. Certain nodes may have more critical data than others and may require the assurance of data validation. When you identify the nodes that require data validation, you can choose to have only the user's data validated or all the data validated. Tivoli Storage Manager validates both the file data and the file metadata when you choose to validate all data. See "Validating a Node's Data During a Client Session" on page 316.

When you enable data validation for a server-to-server exchange or between a storage agent and server, the server must validate all data. You can enable data validation by using the DEFINE SERVER or UPDATE SERVER command. For a server-to-server exchange, see "Using Virtual Volumes to Store Data on Another Server" on page 469 for more information. For data that is exchanged between a storage agent and the server, refer to the *Tivoli Storage Manager Managed System for SAN Storage Agent User's Guide* for the storage agent's operating system.

If the network is fairly stable but your site is perhaps using new hardware devices, you may decide to only enable data validation for storage pools. When the server sends data to the storage pool, the server generates cyclic redundancy checking, and stores the CRC value with the data. The server validates the CRC value when the server audits the volume. Later, you may decide that data validation for storage pools is no longer required after the devices prove to be stable. Refer to "Auditing a Storage Pool Volume" on page 529 for more information on data validation for storage pools.

Performance Considerations

Consider the impact on performance when you decide whether data validation is necessary for storage pools. Data validation impacts performance because the server requires additional CPU overhead to calculate and compare CRC values. This method of validation is independent of validating data during a client session with the server. When you choose to validate storage pool data, there is no performance impact on the client.

If you enable CRC for storage pools on devices that later prove to be stable, you can increase performance by updating the storage pool definition to disable data validation.

Managing Storage Pool Data Validation

The AUDIT VOLUME command has additional parameters that allow you to specify an audit for data written to volumes within a range of days, or to run an audit for a given storage pool.

You can manage when the validation of data in storage pools occurs by scheduling the audit volume operation. You can choose a method suitable to your environment, for example:

- The administrator can select volumes at random to audit. A random selection does not require significant resources or cause much contention for server resources but can provide assurance that the data is valid.
- Schedule a daily audit of all volumes written in the last day. This method validates data written to a given storage pool on a daily basis.
- Audit volumes in storage pools only for client nodes that are considered to be critical users.

Auditing a Volume in a Disk Storage Pool

To audit disk volume /dev/vol1 and have only summary messages sent to the activity log and server console, enter:

```
audit volume /dev/vol1 quiet=yes
```

The audit volume process is run in the background and the server returns the following message:

```
ANR2313I Audit Volume NOFIX process started for volume /dev/vol1
(process id 4).
```

To view the status of the audit volume process, enter:

```
query process
```

The following figure displays an example of the audit volume process report.

```
 Process Process Description      Status
  Number
-------- ------------------------ ---------------------------------------------
       4 Audit Volume             Storage Pool BACKUPPOOL, Volume
         (Inspect Only)            /dev/vol1, Files Processed: 680,
                                   Irretrievable Files Found: 0, Partial Files
                                   Skipped: 0
```

To display the results of a volume audit after it has completed, you can issue the QUERY ACTLOG command.

Auditing Multiple Volumes in a Sequential Access Storage Pool

When you audit a sequential storage volume containing files that span multiple volumes, the server selects all associated volumes. The server begins the audit process with the first volume on which the first file resides. For example, Figure 84 on page 536 on page 536 shows five volumes defined to ENGBACK2. In this example, File A spans VOL1 and VOL2, and File D spans VOL2, VOL3, VOL4, and VOL5.

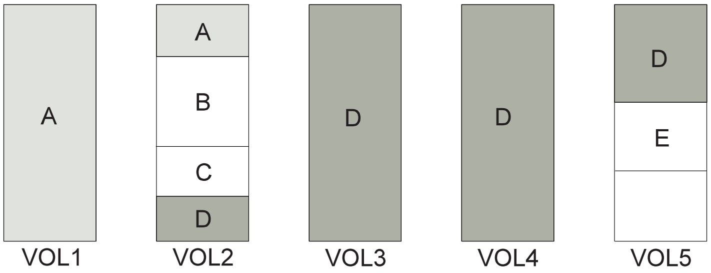


Figure 84. Tape Volumes with Files A, B, C, D, and E

If you request that the server audit volume VOL3, the server first accesses volume VOL2, because File D begins at VOL2. When volume VOL2 is accessed, the server *only* audits File D. It does not audit the other files on this volume.

Because File D spans multiple volumes, the server accesses volumes VOL2, VOL3, VOL4, and VOL5 to ensure that there are no inconsistencies between the database and the storage pool volumes.

For volumes that require manual mount and dismount operations, the audit process can require significant manual intervention.

Auditing a Single Volume in a Sequential Access Storage Pool

To audit a single volume in a sequential storage pool, request that the server skip any files that span multiple volumes. This option is useful when the volume you want to audit contains part of a file, the rest of which resides on a different, damaged volume.

For example, to audit only volume VOL5 in the example in Figure 84 and have the server fix any inconsistencies found between the database and the storage volume, enter:

```
audit volume vol5 fix=yes skippartial=yes
```

Auditing Volumes by Date Written

You can limit the audit to volumes that were written in a certain time range. For example, to audit the volumes in storage pool BKPOOL1 for volumes written from March 20, 2002 to March 22, 2002.

```
audit volume stgpool=bkppool1 fromdate=032002 todate=032202
```

The server audits all volumes that were written to starting at 12:00:01 a.m. on March 20 and ending at 11:59:59 p.m. on March 22, 2002.

When you use the parameters FROMDATE, TODATE, or both, the server limits the audit to only the sequential media volumes that meet the date criteria, and automatically includes all online disk volumes. When you include the STGPOOL parameter you limit the number of volumes that may include disk volumes.

Auditing Volumes in a Specific Storage Pool

You can limit the audit to volumes in a specified storage pool. For example, you can audit the volumes in storage pool BKPOOL1 by issuing the following command:

```
audit volume stgpool=bkppool1
```

Defining a Schedule to Audit Volumes on a Regular Basis

You can define administrative schedules to have the server audit volumes on a regular basis. The following example can be used if your critical users store data in storage pool STPOOL3 and you want all volumes in the storage pool audited every 2 days.

```
define schedule crcstg1 type=administrative
cmd='audit volume stgpool=STPOOL3' active=yes starttime=21:00 period=2
```

This command will audit all volumes in the STPOOL3 storage pool every two days. The audit operation will begin at 9:00 p.m.

Correcting Damaged Files

A data error, which results in a file being unreadable, can be caused by such things as a tape deteriorating or being overwritten or by a drive needing cleaning. If a data error is detected when a client tries to restore, retrieve, or recall a file or during a volume audit, the file is marked as damaged. If the same file is stored in other copy storage pools, the status of those file copies is not changed.

If a client tries to access a damaged file and an undamaged copy is available on an onsite copy storage pool volume, the server sends the user the undamaged copy.

Files that are marked as damaged cannot be:

- Restored, retrieved, or recalled
- Moved by migration, reclamation, or the MOVE DATA command
- Backed up during a BACKUP STGPOOL operation if the primary file is damaged
- Restored during a RESTORE STGPOOL or RESTORE VOLUME operation if the backup copy in a copy storage pool is damaged

Maintaining the Integrity of Files

To maintain the data integrity of user files, you can:

1. Detect damaged files before the users do.

   The AUDIT VOLUME command marks a file as damaged if a read error is detected for the file. If an undamaged copy is in an onsite copy storage pool, it is used to provide client access to the file.
2. Reset the damaged status of files if the error that caused the change to damaged status was temporary.

   You can use the AUDIT VOLUME command to correct situations when files are marked damaged due to a temporary hardware problem, such as a dirty tape head. The server resets the damaged status of files if the volume in which the files are stored is audited and no read errors are detected.
3. Correct files that are marked as damaged.

If a primary file copy is marked as damaged and a usable copy exists in a copy storage pool, the primary file can be corrected using the RESTORE VOLUME or RESTORE STGPOOL command. For an example, see "Restoring Damaged Files".

4. Regularly run commands to identify files that are marked as damaged:
   - The RESTORE STGPOOL command displays the name of each volume in the restored storage pool that contains one or more damaged primary files. Use this command with the preview option to identify primary volumes with damaged files without actually performing the restore operation.
   - The QUERY CONTENT command with the DAMAGED option lets you display damaged files on a specific volume.

For an example of how to use these commands, see "Restoring Damaged Files".

Restoring Damaged Files

If you use copy storage pools, you can restore damaged client files. You can also check storage pools for damaged files and restore the files. This section explains how to restore damaged files based on the scenario in "Example: Simple Hierarchy with One Copy Storage Pool" on page 510.

If a client tries to access a file stored in TAPEPOOL and a read error occurs, the file in TAPEPOOL is automatically marked as damaged. Future accesses to the file automatically use the copy in COPYPOOL as long as the copy in TAPEPOOL is marked as damaged.

To restore any *damaged* files in TAPEPOOL, you can define a schedule that issues the following command periodically:

```
restore stgpool tapepool
```

You can check for and replace any files that develop data-integrity problems in TAPEPOOL or in COPYPOOL. For example, every three months, query the volumes in TAPEPOOL and COPYPOOL by entering the following commands:

```
query volume stgpool=tapepool
```

```
query volume stgpool=copypool
```

Then issue the following command for each volume in TAPEPOOL and COPYPOOL:

```
audit volume <volname> fix=yes
```

If a read error occurs on a file in TAPEPOOL, that file is marked *damaged* and an error message is produced. If a read error occurs on file in COPYPOOL, that file is deleted and a message is produced.

Restore *damaged* primary files by entering:

```
restore stgpool tapepool
```

Finally, create new copies in COPYPOOL by entering:

```
backup stgpool tapepool copypool
```

Backup and Recovery Scenarios

This section presents scenarios for protecting and recovering a Tivoli Storage Manager server. You can modify the procedures to meet your needs.

DRM can help you track your onsite and offsite volumes and query the server and generate a current, detailed disaster recovery plan for your installation.

These scenarios assume a storage hierarchy consisting of:

- The default random access storage pools (BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL)
- TAPEPOOL, a tape storage pool

Protecting Your Database and Storage Pool

A company's standard procedures include the following:

- Perform reclamation of its copy storage pool, once a week. Reclamation for the copy storage pools is turned off at other times.

  Note: In a copy storage pool definition, the REUSEDELAY parameter delays volumes from being returned to scratch or being reused. Set the value high enough to ensure that the database can be restored to an earlier point in time and that database references to files in the storage pool are valid. For example, to retain database backups for seven days and, therefore, sets REUSEDELAY to 7.
- Back up its storage pools every night.
- Perform a full backup of the database once a week and incremental backups on the other days.
- Ship the database and copy storage pool volumes to an offsite location every day.

To protect client data, the administrator does the following:

1. Creates a copy storage pool named DISASTER-RECOVERY. Only scratch tapes are used, and the maximum number of scratch volumes is set to 100. The copy storage pool is defined by entering:

   ```
   define stgpool disaster-recovery tapeclass pooltype=copy
   maxscratch=100
   ```
2. Performs the first backup of the primary storage pools.

   Note: The first backup of a primary storage pool is a full backup and, depending on the size of the storage pool, could take a long time.
3. Defines schedules for the following daily operations:
   a. Incremental backups of the primary storage pools each night by issuing:

      ```
      backup stgpool backuppool disaster-recovery maxprocess=2

      backup stgpool archivepool disaster-recovery maxprocess=2

      backup stgpool spacemgpool disaster-recovery maxprocess=2

      backup stgpool tapepool disaster-recovery maxprocess=2
      ```

   These commands use multiple, parallel processes to perform an incremental backup of each primary storage pool to the copy pool. Only those files for which a copy does not already exist in the copy pool are backed up.

   Note: Migration should be turned off during the rest of the day. You could add a schedule to migrate from disk to tape at this point. In this way, the backups are done while the files are still on disk.

   b. Change the access mode to OFFSITE for volumes that have read-write or read-only access, are onsite, and are at least partially filled. This is done by entering:

      ```
      update volume * access=offsite location='vault site info'
      wherestgpool=disaster-recovery whereaccess=readwrite,readonly
      wherestatus=filling,full
      ```

   c. Back up the database by entering:

      ```
      backup db type=incremental devclass=tapeclass scratch=yes
      ```

4. Does the following operations nightly after the scheduled operations have completed:
   a. Backs up the volume history and device configuration files. If they have changed, back up the server options files and the database and recovery log setup information.
   b. Moves the volumes marked offsite, the database backup volumes, volume history files, device configuration files, server options files and the database and recovery log setup information to the offsite location.
   c. Identifies offsite volumes that should be returned onsite by using the QUERY VOLUME command:

      ```
      query volume stgpool=disaster-recovery access=offsite status=empty
      ```

      These volumes, which have become empty through expiration, reclamation, and file space deletion, have waited the delay time specified by the REUSEDELAY parameter. The administrator periodically returns outdated backup database volumes. These volumes are displayed with the QUERY VOLHISTORY command and can be released for reuse with the DELETE VOLHISTORY command.
5. Brings the volumes identified in step 4c onsite and updates their access to read-write.

Recovering to a Point-in-Time from a Disaster

In this scenario, the processor on which Tivoli Storage Manager resides, the database, and all onsite storage pool volumes are destroyed by fire. An administrator restores the server to the point-in-time of the last backup. You can use either full and incremental backups or snapshot database backups to restore a database to a point-in-time.

DRM can help you do these steps.

Do the following:

1. Install Tivoli Storage Manager on the replacement processor with the same server options and the same size database and recovery log as on the destroyed system. For example, to initialize the database and recovery log, enter:

   ```
   dsmserv format 1 log1 1 dbvol1
   ```
2. Move the latest backup and all of the DISASTER-RECOVERY volumes onsite from the offsite location.

   Note: Do not change the access mode of these volumes until after you have completed step 7.
3. If a current, undamaged volume history file exists, save it.
4. Restore the volume history and device configuration files, the server options and the database and recovery log setup. For example, the recovery site might require different device class, library, and drive definitions. For more information, see "Updating the Device Configuration File" on page 518.
5. Restore the database from the latest backup level by issuing the DSMSERV RESTORE DB utility (see "Recovering Your Server Using Database and Storage Pool Backups" on page 520).
6. Change the access mode of all the existing primary storage pool volumes in the damaged storage pools to DESTROYED by entering:

   ```
   update volume * access=destroyed wherestgpool=backuppool
   update volume * access=destroyed wherestgpool=archivepool
   update volume * access=destroyed wherestgpool=spacemgpool
   update volume * access=destroyed wherestgpool=tapepool
   ```
7. Issue the QUERY VOLUME command to identify any volumes in the DISASTER-RECOVERY storage pool that were onsite at the time of the disaster. Any volumes that were onsite would have been destroyed in the disaster and could not be used for restore processing. Delete each of these volumes from the database by using the DELETE VOLUME command with the DISCARDDATA option. Any files backed up to these volumes cannot be restored.
8. Change the access mode of the remaining volumes in the DISASTER-RECOVERY pool to READWRITE by entering:

   ```
   update volume * access=readwrite wherestgpool=disaster-recovery
   ```

   Note: At this point, clients can access files. If a client tries to access a file that was stored on a destroyed volume, the retrieval request goes to the copy storage pool. In this way, clients can access their files without waiting for the primary storage pool to be restored. When you update volumes brought from offsite to change their access, you greatly speed recovery time.
9. Define new volumes in the primary storage pool so the files on the damaged volumes can be restored to the new volumes. The new volumes also let clients backup, archive, or migrate files to the server. You do not need to perform this step if you use only scratch volumes in the storage pool.
10. Restore files in the primary storage pool from the copies located in the DISASTER-RECOVERY pool by entering:

    ```
    restore stgpool backuppool maxprocess=2
    restore stgpool archivepool maxprocess=2
    restore stgpool spacemgpool maxprocess=2
    restore stgpool tapepool maxprocess=2
    ```

These commands use multiple parallel processes to restore files to primary storage pools. After all the files have been restored for a destroyed volume, that volume is automatically deleted from the database. See “When a Storage Pool Restoration Is Incomplete” on page 527 for what to do if one or more volumes cannot be fully restored.

11. To ensure against another loss of data, immediately back up all storage volumes and the database. Then resume normal activity, including weekly disaster backups and movement of data to the offsite location.

Point-in-Time Restores In a Shared Library Environment

A point-in-time restore for a library manager server or a library client server requires additional steps to ensure the consistency of the volume inventories of the affected servers. This section describes the procedures for the two possible scenarios.

Point-in-Time Restore of a Library Manager Server: A point-in-time restore of a library manager server could create inconsistencies between the volume inventories of the library manager and library client servers. The restore removes all library client server transactions that occurred after the point in time from the volume inventory of the library manager server. The volume inventory of the library client server, however, still contains those transactions. New transactions could then be written to these volumes, resulting in a loss of client data. To prevent this problem, do the following after the restore:

1. Halt further transactions on the library manager server: Disable all schedules, migration and reclamations on the library client and library manager servers.
2. Audit all libraries on all library client servers. The audits will re-enter those volume transactions that were removed by the restore on the library manager server. You should audit the library clients from the oldest to the newest servers. Use the volume history file from the library client and library manager servers to resolve any conflicts.
3. Delete the volumes from the library clients that do not own the volumes.
4. Resume transactions by enabling all schedules, migration, and reclamations on the library client and library manager servers.

Point-in-Time Restore of a Library Client Server: A point-in-time restore of a library client server could cause volumes to be removed from the volume inventory of a library client server and later overwritten. If a library client server acquired scratch volumes after the point-in-time to which the server is restored, these volumes would be set to private in the volume inventories of the library client and library manager servers. After the restore, the volume inventory of the library client server can be regressed to a point-in-time before the volumes were acquired, thus removing them from the inventory. These volumes would still exist in the volume inventory of the library manager server as private volumes owned by the client.

The restored volume inventory of the library client server and the volume inventory of the library manager server would be inconsistent. The volume inventory of the library client server must be synchronized with the volume inventory of the library manager server in order to return those volumes to scratch and enable them to be overwritten. To synchronize the inventories, do the following:

1. Audit the library on the library client server to synchronize the volume inventories of the library client and library manager servers.
2. To resolve any remaining volume ownership concerns, refer to the volume history and issue the UPDATE VOLUME command as needed.

Recovering a Lost or Damaged Storage Pool Volume

If a company makes the preparations described in “Protecting Your Database and Storage Pool” on page 539, it can recover from a media loss. In the following scenario, an operator inadvertently destroys a tape volume (DSM087) belonging to the TAPEPOOL storage pool. An administrator performs the following actions to recover the data stored on the destroyed volume by using the offsite copy storage pool:

1. Determine the copy pool volumes that contain the backup copies of the files that were stored on the volume that was destroyed by entering:

   ```
   restore volume dsm087 preview=yes
   ```

   This command produces a list of offsite volumes that contain the backed up copies of the files that were on tape volume DSM087.
2. Set the access mode of the copy volumes identified as UNAVAILABLE to prevent reclamation.

   Note: This precaution prevents the movement of files stored on these volumes until volume DSM087 is restored.
3. Bring the identified volumes to the onsite location and set their access mode to READONLY to prevent accidental writes. If these offsite volumes are being used in an automated library, the volumes must be checked into the library when they are brought back onsite.
4. Restore the destroyed files by entering:

   ```
   restore volume dsm087
   ```

   This command sets the access mode of DSM087 to DESTROYED and attempts to restore all the files that were stored on volume DSM087. The files are not actually restored to volume DSM087, but to another volume in the TAPEPOOL storage pool. All references to the files on DSM087 are deleted from the database and the volume itself is deleted from the database.
5. Set the access mode of the volumes used to restore DSM087 to OFFSITE using the UPDATE VOLUME command.
6. Set the access mode of the restored volumes, that are now onsite, to READWRITE.
7. Return the volumes to the offsite location. If the offsite volumes used for the restoration were checked into an automated library, these volumes must be checked out of the automated library when the restoration process is complete.

Chapter 23. Using Tivoli Disaster Recovery Manager

You can use Tivoli Disaster Recovery Manager (DRM) product to do any one or all the following:

- Prepare a disaster recovery plan that can help you to recover your applications in the case of a disaster. You can recover at an alternate site, on replacement computer hardware, and with people who are not familiar with the applications.
- Manage your offsite recovery media.
- Store your client recovery information.

You can also use the disaster recovery plan for audits to certify the recoverability of the server.

Note: DRM is a separate program product that must be licensed before using. For more information, see "Registering Licensed Features" on page 350.

Before using this chapter, you should be familiar with "Chapter 22. Protecting and Recovering Your Server" on page 499.

This chapter contains the following sections:

Tasks:
"Specifying Defaults for Tivoli Disaster Recovery Manager" on page 546
"Specifying Recovery Instructions for Your Site" on page 550
"Specifying Information About Your Server and Client Node Machines" on page 551
"Specifying Recovery Media for Client Machines" on page 554
"Creating and Storing the Disaster Recovery Plan" on page 554
"Managing Disaster Recovery Plan Files Stored on Target Servers" on page 556
"Moving Backup Media" on page 558
"Summary of Tivoli Disaster Recovery Manager Daily Tasks" on page 563
"Staying Prepared for a Disaster" on page 564
"Recovering From a Disaster" on page 565
Disaster Recovery Reference:
"Tivoli Disaster Recovery Manager Checklist" on page 572
"The Disaster Recovery Plan File" on page 575

In this chapter, most examples illustrate how to perform tasks by using a Tivoli Storage Manager command-line interface. For information about the commands, see *Administrator's Reference*, or issue the HELP command from the command line of a Tivoli Storage Manager administrative client.

Tivoli Storage Manager tasks can also be performed from the administrative Web interface. For more information about using the administrative interface, see *Quick Start*.

Task	**Required Privilege Class**
All DRM tasks unless otherwise noted.	System

© Copyright IBM Corp. 1993, 2002

Note: The default installation directories changed for Tivoli Storage Manager. If you created a recovery plan file with ADSM Version 3 Release 1, some names in that file may no longer be valid. After installing Tivoli Storage Manager, immediately back up your storage pools and database and create a new recovery plan file.

You can use a recovery plan file and database backup that were created on an ADSM Version 3 Release 1 server to restore a Tivoli Storage Manager server. After the restore is complete, start the server with the following command:

```
dsmserv upgradedb
```

Use the UPGRADEDB parameter only for the initial startup.

To recover from a disaster, you must know the location of offsite recovery media. DRM helps you to determine which volumes to move offsite and back onsite and, tracks the location of the volumes.

Specifying Defaults for Tivoli Disaster Recovery Manager

DRM provides default settings for the preparation of the recovery plan file and for the management of offsite recovery media. However, you can override these default settings. To query the settings, issue the following command:

```
query drmstatus
```

The output will be similar to the following:

```
            Recovery Plan Prefix: /u/recovery/plans/rpp
        Plan Instructions Prefix: /u/recovery/plans/source/
      Replacement Volume Postfix: @
           Primary Storage Pools: PRIM1 PRIM2
              Copy Storage Pools: COPY*
     Not Mountable Location Name: Local
                    Courier Name: Joe's Courier Service
                 Vault Site Name: Ironvault, D. Lastname, 1-000-000-0000
DB Backup Series Expiration Days: 30 Day(s)
                    Check Label?: Yes
       Process FILE Device Type?: No
               Command File Name: /drm/orm/exec.cmds
```

Specifying Defaults for the Disaster Recovery Plan File

The following table describes how to set defaults for the disaster recovery plan file.

Table 33. Defaults for the Disaster Recovery Plan File

Primary storage pools to be processed	When the recovery plan file is generated, you can limit processing to specified pools. The recovery plan file will not include recovery information and commands for storage pools with a data format of NETAPPDUMP. **The default at installation:** All primary storage pools. **To change the default:** SET DRMPRIMSTGPOOL For example, to specify that only the primary storage pools named PRIM1 and PRIM2 are to be processed, enter: `set drmprimstgpool prim1,prim2` **Note:** To remove all previously specified primary storage pool names and thus select all primary storage pools for processing, specify a null string ("") in SET DRMPRIMSTGPOOL. **To override the default:** Specify primary storage pool names in the PREPARE command
Copy storage pools to be processed	When the recovery plan file is generated, you can limit processing to specified pools. **The default at installation:** All copy storage pools. **To change the default:** SET DRMCOPYSTGPOOL For example, to specify that only the copy storage pools named COPY1 and COPY2 are to be processed, enter: `set drmcopystgpool copy1,copy2` **Notes:** 1. To remove any specified primary storage pool names, and thus select all primary storage pools, specify a null string ("") in SET DRMCOPYSTGPOOL. 2. If you specify both primary and copy storage pools, the specified copy storage pools should be those used to back up the specified primary storage pools. **To override the default:** Specify copy storage pool names in the PREPARE command
Identifier for replacement volume names	To restore a primary storage pool volume, mark the original volume *destroyed* and create a replacement volume having a unique name. You can specify a character to be appended to the name of the original volume in order to create a name for the replacement volume. This character can help you find the replacement volume names in the disaster recovery plan. For details about the restore process, see "How Restore Processing Works" on page 500. **The default identifier at installation:** @ **To change the default:** SET DRMPLANVPOSTFIX For example, to use the character r, enter: `set drmplanvpostfix r`

Table 33. Defaults for the Disaster Recovery Plan File (continued)

Recovery instructions prefix	You can specify a prefix for the names of the recovery instructions source files in the recovery plan file.
	The default at installation: For a description of how DRM determines the default prefix, see the INSTRPREFIX parameter of the PREPARE command section in the *Administrator's Reference* or enter HELP PREPARE from administrative client command line.
	To set a default: SET DRMINSTRPREFIX
	For example, to specify the prefix as */u/recovery/plans/rpp*, enter:
	`set drminstrprefix /u/recovery/plans/rpp`
	The disaster recovery plan file will include, for example, the following file:
	`/u/recovery/plans/rpp.RECOVERY.INSTRUCTIONS.GENERAL`
	To override the default: The INSTRPREFIX parameter with the PREPARE command

Prefix for the recovery plan file	You can specify a prefix to the path name of the recovery plan file. DRM uses this prefix to identify the location of the recovery plan file and to generate the macros and script file names included in the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE and RECOVERY.SCRIPT.NORMAL.MODERECOVERY.DRMODE and RECOVERY.NMODE stanzas.
	The default at installation: For a description of how DRM determines the default prefix, see the PLANPREFIX parameter of the PREPARE command section in the *Administrator's Reference* or enter HELP PREPARE from administrative client command line.
	To change the default: SET DRMPLANPREFIX
	For example, to specify the prefix as */u/server/recoveryplans/*, enter:
	`set drmplanprefix /u/server/recoveryplans/`
	The disaster recovery plan file name created by PREPARE processing will be in the following format:
	`/u/server/recoveryplans/20000603.013030`
	To override the default: The PLANPREFIX parameter with the PREPARE command

The disaster recovery plan expiration period	You can set the numbers of days after creation that a disaster recovery plan file stored on a target server expires. After the number of days has elapsed, all recovery plan files that meet both of the following conditions are eligible for expiration: • The last recovery plan associated with the database series is older than the set number of days. • The recovery plan file is not associated with the most recent backup series. **The default at installation:** 60 days **To change the default:** SET DRMRPFEXPIREDAYS For example, to change the time to 90 days, enter: `set drmrpfexpiredays 90`

Specifying Defaults for Offsite Recovery Media Management

The following table describes how to set defaults for offsite recovery media management.

Copy storage pool volumes to be processed	MOVE DRMEDIA and QUERY DRMEDIA can process copy storage pool volumes in the MOUNTABLE state. You can limit processing to specified copy storage pools. **The default at installation:** All copy storage pool volumes in the MOUNTABLE state **To change the default:** SET DRMCOPYSTGPOOL **To override the default:** COPYSTGPOOL parameter on MOVE DRMEDIA or QUERY DRMEDIA
Executable commands file name	You can use MOVE DRMEDIA or QUERY DRMEDIA to generate executable commands and store them in a file. **The default file name at installation:** None **To set a default:** SET DRMCMDFILENAME. For example, to set the file name as */drm/orm/exec.cmds* enter: `set drmcmdfilename /drm/orm/exec.cmds` **To override the default:** CMDFILENAME parameter on MOVE DRMEDIA or QUERY DRMEDIA
Location name for volumes that move to the NOTMOUNTABLE state	MOVE DRMEDIA generates a location name for volumes that move to the NOTMOUNTABLE state. **The default at installation:** NOTMOUNTABLE **To change the default:** SET DRMNOTMOUNTABLENAME For example, to specify a location named LOCAL, enter: `set drmnotmountablename local`
Location name for volumes that move to the COURIER or COURIERRETRIEVE state	MOVE DRMEDIA generates a location name for volumes that are changing from NOTMOUNTABLE to COURIER or from VAULTRETRIEVE to COURIERRETRIEVE. **The default at installation:** COURIER **To change the default:** SET DRMCOURIERNAME For example, to specify a courier named Joe’s Courier Service, enter: `set drmcouriername "Joe's Courier Service"`
Reading labels of checked out volumes	To determine whether DRM reads the sequential media labels of volumes that are checked out with MOVE DRMEDIA. **Note:** This command does not apply to 349X library types. **The default at installation:** DRM reads the volume labels. **To change the default:** SET DRMCHECKLABEL For example, to specify that DRM should not read the volume labels, enter: `set drmchecklabel no`

Expiration period of a database backup series	A database backup series (full plus incremental and snapshot) is eligible for expiration if all of these conditions are true: • The volume state is VAULT or the volume is associated with a device type of SERVER (for virtual volumes). • It is not the most recent database backup series. • The last volume of the series exceeds the expiration value, number of days since the last backup in the series. **The default at installation:** 60 days **To change the default:** SET DRMDBBACKUPEXPIREDAYS For example, to set the expiration value to 30 days, enter: `set drmdbbackupexpiredays 30`
Whether to process backup volumes of the FILE device type	At installation, MOVE DRMEDIA and QUERY DRMEDIA will not process backup volumes that are associated with a device type of FILE. **The default at installation:** Backup volumes of the FILE device type are not processed **To change the default:** SET DRMFILEPROCESS To allow processing, enter: `set drmfileprocess yes`
Vault Name	MOVE DRMEDIA uses the vault name to set the location of volumes that are moving from the COURIER state to the VAULT state **The default at installation:** The vault name is set to VAULT. **To change the default:** SET DRMVAULTNAME For example, to specify the vault name as IRONVAULT, the contact name as J. SMITH, and the telephone number as 1-555-000-0000, enter: `set drmvaultname "Ironvault, J. Smith, 1-555-000-0000"`

Specifying Recovery Instructions for Your Site

The disaster recovery plan includes instructions that you create. Enter your instructions in flat files that have the following names:

- *prefix*.RECOVERY.INSTRUCTIONS.GENERAL
- *prefix*.RECOVERY.INSTRUCTIONS.OFFSITE
- *prefix*.RECOVERY.INSTRUCTIONS.INSTALL
- *prefix*.RECOVERY.INSTRUCTIONS.DATABASE
- *prefix*.RECOVERY.INSTRUCTIONS.STGPOOL

Note: The files created for the recovery instructions must be physical sequential files.

RECOVERY.INSTRUCTIONS.GENERAL

Include information such as administrator names, telephone numbers, and location of passwords. For example:

```
 Recovery Instructions for TSM Server ACMESRV on system ZEUS
 Joe Smith (wk 002-000-1111 hm 002-003-0000):  primary system programmer
 Sally Doe (wk 002-000-1112 hm 002-005-0000):  primary recovery administrator
 Jane Smith (wk 002-000-1113 hm 002-004-0000): responsible manager

Security Considerations:
Joe Smith has the password for the Admin ID ACMEADM. If Joe is unavailable,
you need to either issue SET AUTHENTICATION OFF or define a new
administrative user ID at the replacement TSM server console.
```

RECOVERY.INSTRUCTIONS.OFFSITE
Include information such as the offsite vault location, courier's name, and telephone numbers. For example:

```
Our offsite vault location is Ironvault, Safetown, Az.
The phone number is 1-800-000-0008. You need to contact them directly
to authorize release of the tapes to the courier.
Our courier's name is Fred Harvey.  You can contact him at 1-800-444-0000.
Since our vault is so far away, be sure to give the courier a list
of both the database backup and copy storage pool volumes required. Fred
is committed to returning these volumes to us in less than 12 hours.
```

RECOVERY.INSTRUCTIONS.INSTALL
Include information about restoring the base server system from boot media or, if boot media is unavailable, about server installation and the location of installation volumes. For example:

```
Most likely you will not need to reinstall the TSM server and
administrative clients because we use
mksysb to backup the rootvg volume group, and the TSM server code and
configuration files exist in this group.
However, if you cannot do a mksysb restore of the base server system,
and instead have to start with a fresh AIX build, you may need
to add TSM server code to that AIX system.
The install volume for the TSM server is INS001. If that is lost, you
will need to contact Copy4You Software, at 1-800-000-0000, and obtain
a new copy. Another possibility is the local IBM Branch office
at 555-7777.
```

RECOVERY.INSTRUCTIONS.DATABASE
Include information about how to recover the database and about how much hardware space requirements. For example:

```
You will need to find replacement disk space for the server database. We
have an agreement with Joe Replace that in the event of a disaster, he
will provide us with disk space.
```

RECOVERY.INSTRUCTIONS.STGPOOL
Include information on primary storage pool recovery instructions. For example:

```
Do not worry about the archive storage pools during this disaster recovery.
Focus on migration and backup storage pools.
The most important storage pool is XYZZZZ.
```

Specifying Information About Your Server and Client Node Machines

You need information about your server machine to rebuild its replacement. You also need information about client node machines to rebuild or restore them. Follow this procedure to specify that information and store it in the server database:

Server Machine

1. Specify server machine information:

   Issue the DEFINE MACHINE command. with ADSMSERVER=YES. For example, to define machine MACH22 in building 021, 2nd floor, in room 2929, with a priority of 1, enter:

   ```
   define machine tsm1 adsmserver=yes priority=1
   ```

Client Machines

2. Specify the client node location and business priority:

   Issue the DEFINE MACHINE command. For example, to define machine MACH22 in building 021, 2nd floor, in room 2929, with a priority of 1, enter:

   ```
   define machine mach22 building=021 floor=2 room=2929 priority=1
   ```

3. Associate one or more client nodes with a machine:

   Issue the DEFINE MACHNODEASSOCIATION command. Use this association information to identify client nodes on machines that were destroyed. You should restore the file spaces associated with these nodes. For example, to associate node CAMPBELL with machine MACH22, enter:

   ```
   define machnodeassociation mach22 campbell
   ```

To query machine definitions, issue the QUERY MACHINE command. See the example, in “Client Recovery Scenario” on page 568.

4. To add machine characteristics and recovery instructions to the database, issue the INSERT MACHINE command. You must first query the operating system to identify the characteristics for your client machine. You can add the information manually or use an awk script to do it. A sample program is shipped with DRM.

   - **Add information manually:**

     The following partial output is from a query on an AIX client machine.

     ```
     --1  Host Name: mach22 with 256 MB Memory Card
     ---    256 MB Memory Card
     ---
     --4  Operating System: AIX Version 4 Release 3
     ---
     ---  Hardware Address: 10:00:5x:a8:6a:46
     ```

     Specify characteristics and recovery instructions one line at a time with separate INSERT MACHINE commands:

     – To save the first line (`Host Name: mach22 with 256 MB Memory Card`) as line 1 and to save the fourth line (`Operating System: AIX Version 4 Release 3`) as line 2 for machine MACH22, issue the following commands:

     ```
     insert machine mach22 1 characteristics="Host Name: mach22 with
      256 MB Memory Card"

     insert machine mach22 2 characteristics="Operating System:
      AIX Version 4 Release 3"
     ```

     – To specify recovery instructions for your client machine, issue the following command:

     ```
     insert machine mach22 1 -
       recoveryinstructions="Recover this machine for accounts
        receivable dept."
     ```

   - **Add Information Using an Awk Script**

To help automate the adding of client machine information, a sample awk script named *machchar.awk.smp* is shipped with DRM. The following example shows how to use a local program to add machine characteristics or recovery instructions:

a. The output from the AIX commands *lsdev*, *lsvg*, and *df* is written to the file *clientinfo.txt* on the AIX client machine that backed up data to the server. These commands list the devices, logical volumes by volume group, and file systems.
b. The file, *clientinfo.txt*, is processed by the awk script, which builds a macro of INSERT MACHINE commands (one command for each line in the file).
c. Run the macro to load the data into the database. From an AIX prompt, issue the following commands:

```
echo "devices"                          > clientinfo.txt
lsdev -C | sort -d -f                  >> clientinfo.txt
echo "logical volumes by volume group" >> clientinfo.txt
lsvg -o | lsvg -i -l                   >> clientinfo.txt
echo "file systems"                    >> clientinfo.txt
df                                     >> clientinfo.txt
```

Figure 85 is an example procedure named *machchar* to add machine characteristics. The *machchar.awk.smp* script is shipped with DRM and is located in the */usr/tivoli/tsm/server/bin* directory.

```
# Read machine characteristics from a file and build TSM macro commands
# to insert the information into the machine characteristics table.
# Invoke with:
#   awk -f machchar.awk -v machine=acctrcv filewithinfo

 BEGIN {
       print "delete machine "machine" type=characteri"
       }
       {
       print "insert machine "machine" "NR" characteri=\""$0"\""
       }
 END   {
       }
```

Figure 85. Example of Awk Script File to Insert Machine Characteristics

d. The *machchar.awk* script is then run from an AIX prompt as follows:

```
awk -f machchar.awk -v machine=acctrcv clientinfo.txt >
   clientinfo.mac
```

e. To add the machine characteristics, start an administrative client and run the macro. For example:

```
> dsmadmc -id=xxx -pw=xxx macro clientinfo.mac
```

You can view your machine characteristics by issuing the QUERY MACHINE command with FORMAT=CHARACTERISTICS parameter.

f. To specify recovery instructions for your client machine, use this same awk script process but with the RECOVERYINSTRUCTIONS parameter.

Specifying Recovery Media for Client Machines

Follow these steps to specify the bootable media needed to reinitialize or reinstall an operating system on a client machine and to associate machines with media. You can also associate non-executable media such as application user guides with client machines.

1. Define the bootable media. For example, define the media named TELLERWRKSTNIMAGE which is for AIX Version 4.3, contains the required volumes named AIX001, AIX002, and AIX003, and is located in Building 21.

   ```
   define recoverymedia tellerwrkstnimage type=boot
     volumenames=aix001,aix002,aix003 product="AIX 4.3"
     location="Building 21"
   ```

   You should define the recovery media after a client machine configuration changes. For example, after you have installed a new level of AIX on a client machine and created a bootable image using **mksysb**, issue the DEFINE RECOVERYMEDIA command to define the new **mksysb** volumes.

   To query your recovery media definitions, issue the QUERY RECOVERYMEDIA command with the FORMAT=DETAILED parameter.
2. Associate one or more machines with recovery media. Use the association information to identify the boot media to use in the replacement machines. For example, to associate machine MACH255 with recovery media TELLERWRKSTNIMAGE, issue the following command:

   ```
   define recmedmachassociation tellerwrkstnimage mach255
   ```
3. When the boot media is moved offsite, update its location. For example, to update the location of boot media TELLERWRKSTNIMAGE to the offsite location IRONVAULT, issue the following command:

   ```
   update recoverymedia tellerwrkstnimage location=ironvault
   ```

You can define media that contain softcopy manuals that you would need during recovery. For example, to define a CD-ROM containing the AIX 4.3 manuals that are on volume CD0001, enter:

```
define recoverymedia aix43manuals type=other volumes=cd0001
  description="AIX 4.3 Bookshelf"
```

Creating and Storing the Disaster Recovery Plan

You can create a disaster recovery plan file and store the file locally or on another server.

The recovery plan contains the following information:

- The recovery procedure
- A list of required database and storage pool backup volumes, devices to read those volumes, and database and recovery log space requirements
- Copies of the server options file, device configuration file, and volume history information file
- Commands for performing database recovery and primary storage pool recovery
- Commands for registering licenses
- Instructions that you define
- Machine and recovery media information that you define

For details about the recovery plan file, see "The Disaster Recovery Plan File" on page 575 .

DRM creates one copy of the disaster recovery plan file each time you issue the PREPARE command. You should create multiple copies of the plan for safekeeping. For example, keep copies in print, on diskettes, on NFS-mounted disk space that is located offsite, or on a remote server.

Before creating a disaster recovery plan, back up your storage pools then backup the database. See "Backing Up Storage Pools" on page 507 and "Backing Up the Database" on page 510 for details about these procedures.

If you manually send backup media offsite, see "Moving Backup Volumes Offsite" on page 560. If you use virtual volumes, see "Using Virtual Volumes to Store Data on Another Server" on page 469.

When your backups are both offsite and marked offsite, you can create a disaster recovery plan.

You can use the Tivoli Storage Manager scheduler to periodically run the PREPARE command (see "Chapter 17. Automating Server Operations" on page 367).

Note: DRM creates a plan that assumes that the latest database full plus incremental series would be used to restore the database. However, you may want to use DBSNAPSHOT backups for disaster recovery and retain your full plus incremental backup series on site to recover from possible availability problems. In this case, you must specify the use of DBSNAPSHOT backups in the PREPARE command. For example:

```
prepare source=dbsnapshot
```

Storing the Disaster Recovery Plan Locally

When you create a recovery plan file but do not specify a device class, the file is stored locally in a file system. If you store the file locally, you can specify a storage location. For example, to store the recovery plan file locally in the */u/server/recoveryplans/* directory, enter:

```
prepare planprefix=/u/server/recoveryplans/
```

Recovery plan files that are stored locally are not automatically expired. You should periodically delete down-level recovery plan files manually.

DRM appends to the file name the date and time (yyyymmdd.hhmmss). For example:

```
/u/server/recoveryplans/20000925.120532
```

Storing the Disaster Recovery Plan on a Target Server

When you create a recovery plan file and specify a device class, the file is stored on a target server. Storing recovery plan files on a target server provides the following:

- A central repository on a target server for recovery plan files
- Automatic expiration of plan files
- Query capabilities that display information about recovery plan files and the ability to display the contents of a recovery plan file located on a target server

- Recovery plan file retrieval from a target server

First, set up the source and target servers and define a device class a device type of SERVER (see "Setting Up Source and Target Servers for Virtual Volumes" on page 471 for details). For example, assume a device class named TARGETCLASS is defined on the source server where you create the recovery plan file. Then to create the plan file, enter:

```
prepare devclass=targetclass
```

The recovery plan file is written as an object on the target server, and a volume history record is created on the source server. For more about recovery plan files that are stored on target servers, see "Displaying Information about Recovery Plan Files".

Managing Disaster Recovery Plan Files Stored on Target Servers

The following sections describe how you can view information about disaster recovery plan files stored on a target server and view their contents. It also describes how to direct the contents of a disaster recovery plan file to another file and how to delete volume history records of the recovery plan files.

Displaying Information about Recovery Plan Files

You can display information about recovery plan files from the server that created the files (the source server) or from the server on which the files are stored (the target server):

- **From the source server:** Issue QUERY RPFILE the command with the DEVCLASS parameter that was used on the PREPARE command. Specify the type of database backups that were assumed when the plan was created (either full plus incremental or snapshot). For example, to display a list of all recovery plan files that have been saved for the source server on any target servers and created assuming snapshot database backups, enter:

  ```
  query rpfile devclass=* source=dbsnapshot
  ```

  You can also issue the QUERY VOLHISTORY command to display a list of recovery plan files for the source server. Specify recovery plan files that were created assuming either full plus incremental database backups (TYPE=RPFILE) or database snapshot backups (TYPE=RPFSNAPSHOT). For example:

  ```
  query volhistory type=rpfile
  ```
- **From the target server:** Issue a QUERY RPFILE command that specifies the node name associated with the server or servers that prepared the plan. For example, to display a list of all recovery plan files that have been saved in the target server, enter:

  ```
  query rpfile nodename=*
  ```

Displaying the Contents of a Recovery Plan File

From the server that created the recovery plan file (the source server) or from the server on which the file is stored (the target server), you can display the contents of that file that was saved as on object on the target server. For example,

- **From the source server:** Issue the following command for a recovery plan file created on September 1, 2000 at 4:39 a.m. with the device class TARGETCLASS:

  ```
  query rpfcontent marketing.20000901.043900 devclass=targetclass
  ```

- **From the target server:** Issue the following command for a recovery plan file created on August 31,2000 at 4:50 a.m. on a source server named MARKETING whose node name is BRANCH8:

```
query rpfcontent marketing.20000831.045000 nodename=branch8
```

Notes:

1. You cannot issue these commands from a server console.
2. An output delay can occur when the plan file is located on tape.

See "The Disaster Recovery Plan File" on page 575 for an example of the contents of a recovery plan file.

Restoring a Recovery Plan File

To restore a recovery plan file, use the QUERY RPFCONTENT command and direct the output to a file. You can issue the command from the server that created the files (the source server) or from the server on which the files are stored (the target server). To see a list of recovery plan file names, issue the QUERY RPFILE command.

For example, a recovery plan file named *marketing.20000831.045000* was created using the device class of TARGETCLASS and on a source server whose node name at the target server is BRANCH8. You want to restore the file and direct the output to *rpf.out*:

- **From the source server:** Enter,

```
query rpfcontent marketing.20000831.045000
  devclass=targetclass > rpf.out
```

- **From the target server:** Enter,

```
query rpfcontent marketing.20000831.045000
  nodename=branch8 > rpf.out
```

To display a list of recovery plan files, use the QUERY RPFILE command. See "Displaying Information about Recovery Plan Files" on page 556 for more information.

Expiring Recovery Plan Files Automatically

You can set DRM to expire recovery plan files a certain number of days after they are created. To set up expiration, issue the SET DRMRPFEXPIREDAYS command. The default value is 60 days. For example, to change the time to 90 days, enter:

```
set drmrpfexpiredays 90
```

All recovery plan files that meet the criteria are eligible for expiration if both of the following conditions exist:

- The last recovery plan file of the series is over 90 days old.
- The recovery plan file is not associated with the most recent backup series. A backup series consists of a full database backup and all incremental backups that apply to that full backup. Another series begins with the next full backup of the database.

Expiration applies to plan files based on both full plus incremental and snapshot database backups.

Deleting Recovery Plan Files Manually

You can delete volume history records containing information about recovery plan file objects. When the records are deleted from the source server and the grace period is reached, the objects are deleted from the target server.

Note: The record for the latest recovery plan file is not deleted.

For example, to delete records for recovery plan files that were created on or before 08/30/2000 and assuming full plus incremental database backup series, enter:

```
delete volhistory type=rpfile todate=08/30/2000
```

To limit the operation to recovery plan files that were created assuming database snapshot backups, specify TYPE=RPFSNAPSHOT.

Moving Backup Media

To recover from a disaster you will need database backup volumes and copy storage pool volumes. To prepare for a disaster, you will need perform the following daily tasks:

1. Move new backup media offsite and update the database with their locations. See "Moving Backup Volumes Offsite" on page 560 for details.
2. Return expired or reclaimed backup media onsite and update the database with their locations. See "Moving Backup Volumes Onsite" on page 561 for details.

Task	Required Privilege Class
Send backup volumes offsite and back onsite	Unrestricted storage or operator

Offsite recovery media management does not process virtual volumes. To display all virtual copy storage pool and database backup volumes that have their backup objects on the remote target server, issue the following command:

```
query drmedia * wherestate=remote
```

The disaster recovery plan includes backup volume location information and can provide a list of offsite volumes required to restore a server.

The following diagram shows the typical life cycle of the recovery media:

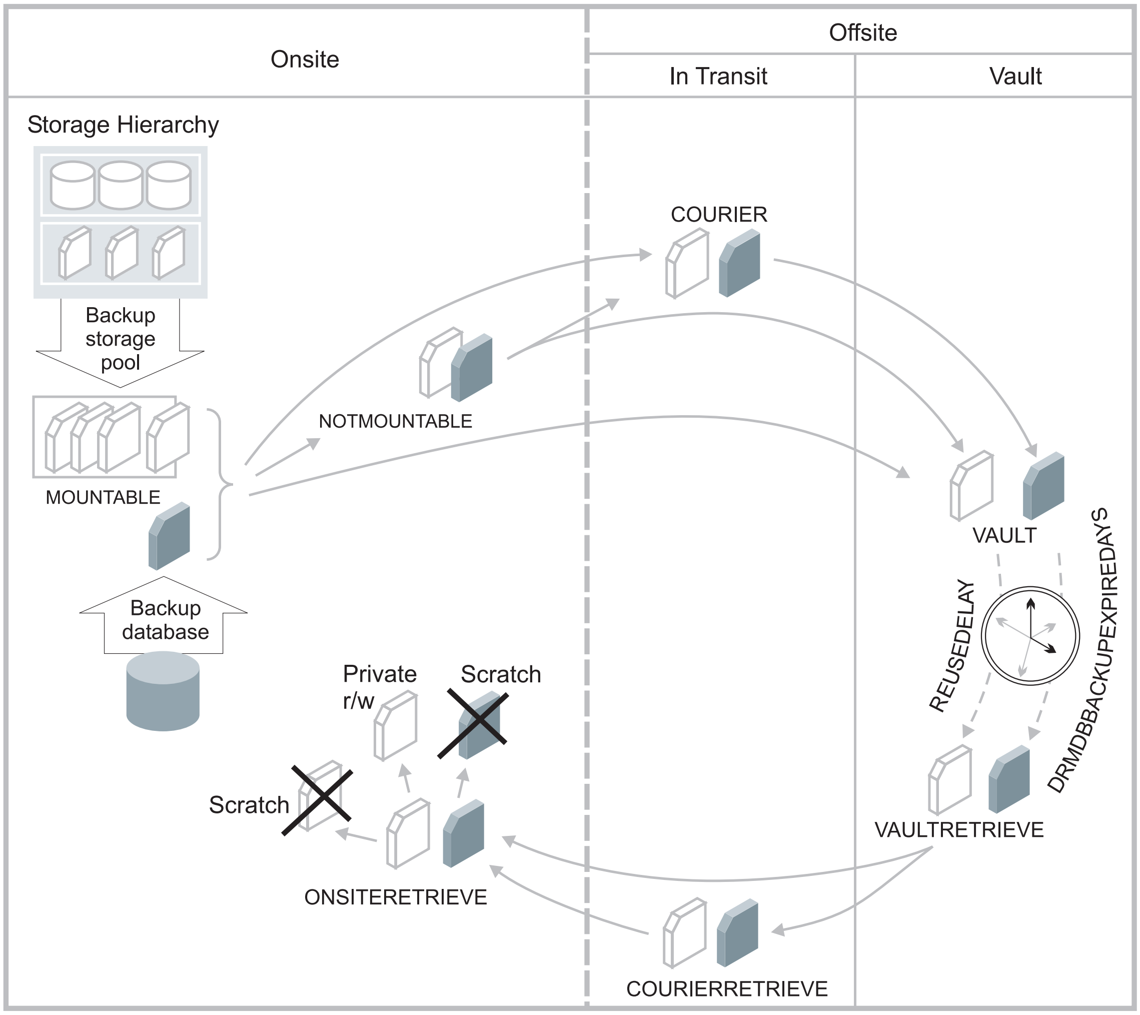


Figure 86. Recovery Media Life Cycle

DRM assigns the following states to volumes. The location of a volume is known at each state.

MOUNTABLE
The volume contains valid data, and TSM can access it.

NOTMOUNTABLE
The volume contains valid data and is onsite, but TSM cannot access it.

COURIER
The volume contains valid data and is in transit to the vault.

VAULT
The volume contains valid data and is at the vault.

VAULTRETRIEVE
The volume no longer contain valid data and are to be returned to the site. For more information on reclamation of offsite copy storage pool volumes, see “Reclamation of Offsite Volumes” on page 193. For information on expiration of database backup volumes, see step 1 on page 561 below.

COURIERRETRIEVE
The volume no longer contain valid data and are in the process of being returned by the courier.

ONSITERETRIEVE

The volume no longer contain valid data and have been moved back to the onsite location. The volume records of database backup and scratch copy storage pool volumes are deleted from the database. For private copy storage pool volumes, the access mode is updated to READWRITE.

Moving Backup Volumes Offsite

After you have created the backup copies of your primary storage pools and database, you can send your backup media offsite. To send media offsite, mark the volumes as unavailable to TSM and give them to the courier. Do the following to identify the database backup and copy storage pool volumes and move them offsite:

1. Identify the copy storage pool and database backup volumes to be moved offsite:

   ```
   query drmedia * wherestate=mountable
   ```

   DRM displays information similar to the following:

   ```
   Volume Name      State            Last Update          Automated
                                     Date/Time            LibName
   ---------------  ---------------  -------------------  -----------------
    TPBK05          Mountable        01/01/2000 12:00:31  LIBRARY
    TPBK99          Mountable        01/01/2000 12:00:32  LIBRARY
    TPBK06          Mountable        01/01/2000 12:01:03  LIBRARY
   ```

2. Indicate the movement of volumes whose current state is MOUNTABLE by issuing the following command:

   ```
   move drmedia * wherestate=mountable
   ```

   For all volumes in the MOUNTABLE state, DRM does the following:

   - Updates the volume state to NOTMOUNTABLE and the volume location according to the SET DRMNOTMOUNTABLENAME. If this command has not been issued, the default location is NOTMOUNTABLE.
   - For a copy storage pool volume, updates the access mode to unavailable.
   - For a volume in an automated library, checks the volume out of the library.

   Notes:

   a. During checkout processing, SCSI libraries request operator intervention. To bypass these requests and eject the cartridges from the library, first issue the following command:

      ```
      move drmedia * wherestate=mountable remove=no
      ```

      Next, access a list of the volumes by issuing the following command:

      ```
      query drmedia wherestate=notmountable
      ```

      From this list identify and remove the cartridges (volumes) from the library.

   b. For the 349X library type, if the number of cartridges to be checked out of the library is greater than the number of slots in the I/O station, you can define a high capacity area in your library. Then use the following command to eject the cartridges to the high capacity area, rather than to the I/O station:

      ```
      move drmedia * wherestate=mountable remove=bulk
      ```

3. Send the volumes to the offsite vault. Issue the following command to have DRM select volumes in the NOTMOUNTABLE state:

   ```
   move drmedia * wherestate=notmountable
   ```

For all volumes in the NOTMOUNTABLE state, DRM updates the volume state to COURIER and the volume location according to the SET DRMCOURIERNAME. If the SET command has not yet been issued, the default location is COURIER. For more information, see “Specifying Defaults for Offsite Recovery Media Management” on page 548

4. When the vault location confirms receipt of the volumes, issue the MOVE DRMEDIA command in the COURIER state. For example:

   ```
   move drmedia * wherestate=courier
   ```

   For all volumes in the COURIER state, DRM updates the volume state to VAULT and the volume location according to the SET DRMVAULTNAME command. If the SET command has not yet been issued, the default location is VAULT. For more information, see “Specifying Defaults for Offsite Recovery Media Management” on page 548.
5. To display a list of volumes that contain valid data at the vault, issue the following command:

   ```
   query drmedia wherestate=vault
   ```

   DRM displays information similar to the following:

   ```
   Volume Name       State          Last Update          Automated
                                    Date/Time            LibName
   ----------------- -------------- -------------------  -----------------
   TAPE0P            Vault          01/05/2000 10:53:20
   TAPE1P            Vault          01/05/2000 10:53:20
   DBT02             Vault          01/05/2000 10:53:20
   TAPE3S            Vault          01/05/2000 10:53:20
   ```
6. If you do not want to step through all the states, you can use the TOSTATE parameter on the MOVE DRMEDIA command to specify the destination state. For example, to transition the volumes from NOTMOUNTABLE state to VAULT state, issue the following command:

   ```
   move drmedia * wherestate=notmountable tostate=vault
   ```

   For all volumes in the NOTMOUNTABLE state, DRM updates the volume state to VAULT and the volume location according to the SET DRMVAULTNAME command. If the SET command has not yet been issued, the default location is VAULT.

See “Staying Prepared for a Disaster” on page 564 for an example that demonstrates sending server backup volumes offsite using MOVE DRMEDIA and QUERY DRMEDIA commands.

Moving Backup Volumes Onsite

Use the following procedure to expire the non-virtual database backup volumes and return the volumes back onsite for reuse or disposal.

1. To specify the number of days before a database backup series is expired, issue the SET DRMDBBACKUPEXPIREDAYS command. To ensure that the database can be returned to an earlier level and database references to files in the copy storage pool are still valid, specify the same value for the REUSEDELAY parameter in your copy storage pool definition.

   The following example sets the number of days to 30.

   ```
   set drmdbbackupexpiredays 30
   ```

A database backup volume is considered eligible for expiration if all of the following conditions are true:

- The age of the last volume of the series has exceeded the expiration value. This value is the number of days since the last backup in the series. At installation, the expiration value is 60 days. To override this value, issue the SET DRMDBBACKUPEXPIREDAYS command.
- For volumes that are not virtual volumes, all volumes in the series are in the VAULT state.
- The volume is not part of the most recent database backup series.

Note: Database backup volumes that are virtual volumes are removed during expiration processing. This processing is started manually by issuing the EXPIRE INVENTORY command or automatically through the EXPINTERVAL option setting specified in the server options file.

2. Move a backup volume onsite for reuse or disposal when the volume is reclaimed and:
   - The status for a copy storage pool volume is EMPTY.
   - The database backup series is EXPIRED.

   To determine which volumes to retrieve, issue the following command:

   ```
   query drmedia * wherestate=vaultretrieve
   ```
3. After the vault location acknowledges that the volumes have been given to the courier, issue the following command:

   ```
   move drmedia * wherestate=vaultretrieve
   ```

   The server does the following for all volumes in the VAULTRETRIEVE state:
   - Change the volume state to COURIERRETRIEVE.
   - Update the location of the volume according to what is specified in the SET DRMCOURIERNAME command. For more information, see “Specifying Defaults for Offsite Recovery Media Management” on page 548.
4. When the courier delivers the volumes, acknowledge that the courier has returned the volumes onsite, by issuing:

   ```
   move drmedia * wherestate=courierretrieve
   ```

   The server does the following for all volumes in the COURIERRETRIEVE state:
   - The volumes are now onsite and can be reused or disposed.
   - The database backup volumes are deleted from the volume history table.
   - For scratch copy storage pool volumes, the record in the database is deleted. For private copy storage pool volumes, the access is updated to read/write.
5. If you do not want to step through all the states, you can use the TOSTATE parameter on the MOVE DRMEDIA command to specify the destination state. For example, to transition the volumes from VAULTRETRIEVE state to ONSITERETRIEVE state, issue the following command:

   ```
   move drmedia * wherestate=vaultretrieve tostate=onsiteretrieve
   ```

   The server does the following for all volumes with in the VAULTRETRIEVE state:
   - The volumes are now onsite and can be reused or disposed.
   - The database backup volumes are deleted from the volume history table.
   - For scratch copy storage pool volumes, the record in the database is deleted. For private copy storage pool volumes, the access is updated to read/write.

Summary of Tivoli Disaster Recovery Manager Daily Tasks

This section summarizes the use of DRM during routine operations and during disaster recovery.

Setup

1. License DRM
2. Ensure the device configuration and volume history files exist.
3. Back up the storage pools.
4. Do a full backup the database (for example, a database snapshot backup).
5. Define site-specific server recovery instructions.
6. Describe priority client machines.
7. Generate the disaster recovery plan.

Daily Preparation Operations

Day 1

1. Back up client files.
2. Back up the primary storage pools.
3. Back up the database (for example, a database snapshot backup).
4. Mark the backup volumes as unavailable to TSM.
5. Send the backup volumes and disaster recovery plan file to the vault.
6. Generate the disaster recovery plan.

Day 2

1. Back up client files
2. Back up the primary storage pools.
3. Back up the database (for example, a database snapshot backup).
4. Mark the backup volumes as unavailable to TSM.
5. Send the backup volumes and disaster recovery plan file to the vault.
6. Generate the disaster recovery plan.

Day 3

1. Automatic storage pool reclamation processing occurs.
2. Back up client files.
3. Back up the primary storage pools.
4. Back up the database (for example, a database snapshot backup).
5. Send the backup volumes and a list of expired volumes to be reclaimed to the vault.
6. The vault acknowledges receipt of the volumes sent on the previous day.
7. Generate the disaster recovery plan.

Disaster and Recovery

Day 4

The server and the client machines are destroyed.

1. Restore the server using the latest recovery plan.
2. Identify the top priority client nodes at the disaster site.
3. Restore client machine files from the copy storage pools.
4. Restore the primary storage pools.
5. Move database backup and copy storage pool volumes to the vault.

Daily Operations

Day 5

1. Back up client files.
2. Back up the primary storage pools.
3. Back up the database (for example, a database snapshot backup).

4. Send the backup volumes and a list of expired volumes to be reclaimed to the vault.
5. Generate the disaster recovery plan.

Staying Prepared for a Disaster

This section provides an overview and a scenario of the tasks required to stay prepared for a disaster. The steps are performed by the onsite TSM administrator unless otherwise indicated.

1. Record the following information in the RECOVERY.INSTRUCTIONS stanza source files:
   - Software license numbers
   - Sources of replacement hardware
   - Any recovery steps specific to your installation
2. Store the following information in the database:
   - Server and client node machine information (DEFINE MACHINE, DEFINE MACHINENODE ASSOCIATION, and INSERT MACHINE)
   - The location of the boot recovery media (DEFINE RECOVERYMEDIA)
3. Schedule automatic nightly backups to occur in the following order:
   a. Primary Storage Pools
   b. Database
4. Daily, create a list of the previous night's database and storage pool backup volumes to be sent offsite:

   ```
   query drmedia * wherestate=mountable
   ```

   a. Check the volumes out of the library:

   ```
   move drmedia * wherestate=mountable
   ```

   b. Send the volumes offsite and record that the volumes were given to the courier:

   ```
   move drmedia * wherestate=notmountable
   ```

5. Create a new recovery plan:

   ```
   prepare
   ```

6. Copy the recovery plan file to a diskette to be given to the courier.
7. Create a list of tapes that contain data that is no longer valid and that should be returned to the site:

   ```
   query drmedia * wherestate=vaultretrieve
   ```

8. Give the courier the database and storage pool backup tapes, the recovery plan file diskette, and the list of volumes to be returned from the vault.
9. The courier gives you any tapes that were on the previous day's return from the vault list.

   Update the state of these tapes and check them into the library:

   ```
   move drmedia * wherestate=courierretrieve cmdf=/drm/checkin.libvol
     cmd="checkin libvol libauto &vol  status=scratch"
   ```

   The volume records for the tapes that were in the COURIERRETRIEVE state are deleted from the database. The MOVE DRMEDIA command also generates the CHECKIN LIBVOL command for each tape processed in the file */drm/checkin.libvol*. For example:

   ```
   checkin libvol libauto tape01 status=scratch
   checkin libvol libauto tape02 status=scratch
   ...
   ```

Note: An administrator can run the MACRO command by specifying /drm/checkin.libvol.

```
> dsmadmc -id=xxxxx -pa=yyyyyy MACRO /drm/checkin.libvol
```

10. The courier takes the database and storage pool backup tapes, the recovery plan diskette, and the list of volumes to return from the vault.
11. Call the vault and verify that the backup tapes arrived and are secure, and that the tapes to be returned to the site have been given to the courier.
12. Set the location of the volumes sent to the vault:

    ```
    move drmedia * wherestate=courier
    ```

13. Set the location of the volumes given to the courier by the vault:

    ```
    move drmedia * wherestate=vaultretrieve
    ```

Recovering From a Disaster

This section provides an overview of the tasks involved in recovering the server and clients. It also presents scenarios of both procedures.

Recovering the Server: Here are guidelines for recovering your server:

1. Obtain the latest disaster recovery plan file.
2. Break out the file to view, update, print, or run as macros or scripts (for example, batch programs or batch files).
3. Obtain the backup volumes from the vault.
4. Locate a suitable replacement machine.
5. Restore an AIX image to your replacement machine.
6. Review the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE RECOVERY.SCRIPT.NORMAL.MODE scripts because they are important for restoring the server to a point where clients can be recovered (see "Disaster Recovery Mode Stanza" on page 583).

Recovering the Clients: To recover clients, do the following:

1. Get the following information by querying the recovered database:
    - Client machines that have been defined to TSM, along with their location and restore priority value
    - The location of the boot recovery media
    - Specific recovery instructions for the machine
    - Hardware requirements for the machine
2. With this information restore the client machines.

Server Recovery Scenario

Here is the procedure for a complete recovery of the server after a disaster has destroyed it. In this example virtual volumes are not used. The steps are performed by the onsite administrator unless otherwise indicated.

1. Review the recovery steps described in the RECOVERY.INSTRUCTIONS.GENERAL stanza of the plan.
2. Request the server backup tapes from the offsite vault.
3. Break out the recovery plan file stanzas into multiple files (see "Breaking Out a Disaster Recovery Plan File" on page 575.) These files can be viewed, updated, printed, or run as TSM macros or scripts.
4. Print the RECOVERY.VOLUMES.REQUIRED file. Give the printout to the courier to retrieve the backup volumes.

5. Find a replacement server. The RECOVERY.DEVICES.REQUIRED stanza specifies the device type that is needed to read the backups. The SERVER.REQUIREMENTS stanza specifies the disk space required.
6. Restore an AIX image to the replacement server using a **mksysb** tape. This tape, which includes the TSM server software, is created whenever software updates or configuration changes are made to the AIX system. The tape location should be specified in the RECOVERY.INSTRUCTIONS.INSTALL stanza.

   Restoration from the **mksysb** tapes includes recreating the root volume group, and the file system where the database, recovery log, storage pool and disk volumes are located..
7. Review the TSM macros contained in the recovery plan.

   If, at the time of the disaster, the courier had not picked up the previous night's database and storage pool incremental backup volumes but they were not destroyed, remove the entry for the storage pool backup volumes from the COPYSTGPOOL.VOLUMES.DESTROYED file.
8. If some required storage pool backup volumes could not be retrieved from the vault, remove the volume entries from the COPYSTGPOOL.VOLUMES.AVAILABLE file.
9. If all primary volumes were destroyed, no changes are required to the PRIMARY.VOLUMES script and TSM macro files.
10. Review the device configuration file to ensure that the hardware configuration at the recovery site is the same as the original site. Any differences must be updated in the device configuration file. Examples of configuration changes that require updates to the configuration information are:
    - Different device names
    - Use of a manual library instead of an automated library
    - For automated libraries, the requirement of manually placing the database backup volumes in the automated library and updating the configuration information to identify the element within the library. This allows the server to locate the required database backup volumes.

    For information about updating the device configuration file, see "Updating the Device Configuration File" on page 518.
11. To restore the database to a point where clients can be recovered, invoke the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script file. Enter the script file name at the command prompt. As an alternative, you can use the recovery script as a guide and manually issue the steps.

    The following are some sample steps from a recovery script:
    a. Copy the TSM server options file from the DSMSERV.OPT file to its original location.
    b. Copy the volume history file required by database restore processing from the VOLUME.HISTORY.FILE file to its original location.

       Note: Use this copy of the volume history file unless you have a more recent copy (after the disaster occurred).
    c. Copy the device configuration file required by database restore processing from the DEVICE.CONFIGURATION.FILE file to its original location.
    d. Create the TSM server recovery log and database volumes using DSMFMT.
    e. Issue DSMSERV FORMAT command to format the recovery log and database files.
    f. Issue the DSMSERV RESTORE DB command.

g. Start the server.

h. Register TSM server licenses.

i. Mark copy storage pool volumes retrieved from vault as available.

j. Mark copy storage pool volumes that cannot be obtained as unavailable.

k. Mark primary storage pool volumes as *destroyed*.

Notes:

a. Due to changes in hardware configuration during recovery, you might have to update the device configuration file located in the restored TSM database (see "Updating the Device Configuration File" on page 518).

b. You can mount copy storage pool volumes upon request, check in the volumes in advance, or manually place the volumes in the library and ensure consistency by issuing the AUDIT LIBRARY command.

c. Use the AUDIT LIBRARY command to ensure that the restored TSM database is consistent with the automated library volumes.

12. If client machines are not damaged, invoke the RECOVERY.SCRIPT.NORMAL.MODE script file to restore the server primary storage pools. If client machines are damaged, you may want to delay this action until after all clients are recovered.

    Note: This action is optional because TSM can access the copy storage pool volumes directly to restore client data. Using this feature, you can minimize client recovery time because server primary storage pools do not have to be restored first. However, in this scenario, the client machines were not damaged, so the focus of the administrator is to restore full TSM server operation.

    As an alternative, you can use the recovery script as a guide and manually run each step. The steps run in this script are:

    a. Create replacement primary volumes.

    b. Define the replacement primary volumes to TSM.

    c. Restore the primary storage pools.

13. Collect the database backup and copy storage pool volumes used in the recovery for return to the vault. For these backup volumes to be returned to the vault using the routine MOVE DRMEDIA process, issue the following commands:

    ```
    update volhist TPBK50 devcl=lib8mm ormstate=mountable
    update volhist TPBK51 devcl=lib8mm ormstate=mountable
    ```

    The copy storage pool volumes used in the recovery already have the correct ORMSTATE.

14. Issue the BACKUP DB command to back up the newly restored database.

15. Issue the following command to check the volumes out of the library:

    ```
     move drmedia * wherestate=mountable
    ```

16. Create a list of the volumes to be given to the courier:

    ```
    query drmedia * wherestate=notmountable
    ```

17. Give the volumes to the courier and issue the following command:

    ```
    move drmedia * wherestate=notmountable
    ```

18. Issue the PREPARE command.

Client Recovery Scenario

The following scenario demonstrates the recovery of clients.

1. To view a list of client machines that were lost in building 21 and their restore priority, issue the following command:

   ```
   query machine building=021 format=detailed
   ```

   DRM displays information similar to the following:

   ```
             Machine Name: POLARIS
         Machine Priority: 1
                 Building: 21
                    Floor: 2
                     Room: 1
                  Server?: No
              Description: Payroll
                Node Name: POLARIS
      Recovery Media Name: MKSYSB1
         Characteristics?: Yes
   Recovery Instructions?: Yes
   ```

2. For *each* machine, issue the following commands:

   a. Determine the location of the boot media. For example:

   ```
   query recoverymedia mksysb1
   ```

   The server displays the following information:

   ```
   Recovery Media Name  Volume Names    Location    Machine Name
   -------------------- -----------     ----------  ----------------
   MKSYSB1              vol1 vol2       IRONVAULT    POLARIS
                         vol3
   ```

   b. Determine the machine-specific recovery instructions. For example:

   ```
   query machine polaris format=recoveryinstructions
   ```

   The server displays the following:

   ```
   Recovery Instructions for Polaris.
   Primary Contact:
      Jane Smith (wk 520-000-0000 hm 520-001-0001)
   Secondary Contact:
      John Adams (wk 520-000-0001 hm 520-002-0002)
   ```

   c. Determine the machine hardware requirements. For example:

   ```
   query machine polaris format=characteristics
   ```

   The server displays information similar to the following:

```
devices
aio0         Defined                Asynchronous I/O
bus0         Available 00-00        Microchannel Bus
fd0          Available 00-00-0D-00  Diskette Drive
fda0         Available 00-00-0D     Standard I/O Diskette Adapter
fpa0         Available 00-00        Floating Point Processor
gda0         Available 00-04        Color Graphics Display Adapter
hd1          Defined                Logical volume
hd2          Defined                Logical volume
hd3          Defined                Logical volume
hdisk0       Available 00-01-00-00  400 MB SCSI Disk Drive
hdisk1       Available 00-01-00-40  Other SCSI Disk Drive
hft0         Available              High Function Terminal Subsystem
inet0        Available              Internet Network Extension
ioplanar0    Available 00-00        I/O Planar
kbd0         Defined   00-00-0K-00  United States keyboard
lb0          Available 00-02-00-20  TIVSM Library
lo0          Available              Loopback Network Interface
loglv00      Defined                Logical volume
lp0          Available 00-00-0P-00  IBM 4201 Model 3 Proprinter III
lv03         Defined                Logical volume
lv04         Defined                Logical volume
lvdd         Available              N/A
mem0         Available 00-0B        8 MB Memory Card
mem1         Available 00-0C        16 MB Memory Card
mous0        Defined   00-00-0M-00  3 button mouse
mt0          Available 00-02-00-40  TIVSM Tape Drive
ppa0         Available 00-00-0P     Standard I/O Parallel Port Adapter
pty0         Available              Asynchronous Pseudo-Terminal
rootvg       Defined                Volume group
sa0          Available 00-00-S1     Standard I/O Serial Port 1
sa1          Available 00-00-S2     Standard I/O Serial Port 2
scsi0        Available 00-01        SCSI I/O Controller
scsi1        Available 00-02        SCSI I/O Controller
sio0         Available 00-00        Standard I/O Planar
siokb0       Available 00-00-0K     Keyboard Adapter
sioms0       Available 00-00-0M     Mouse Adapter
siotb0       Available 00-00-0T     Tablet Adapter
sys0         Available 00-00        System Object
sysplanar0   Available 00-00        CPU Planar
sysunit0     Available 00-00        System Unit
tok0         Available 00-03        Token-Ring High-Performance Adapter
tr0          Available              Token Ring Network Interface
tty0         Available 00-00-S1-00  Asynchronous Terminal
tty1         Available 00-00-S2-00  Asynchronous Terminal
usrvice      Defined                Logical volume
veggie2      Defined                Volume group
logical volumes by volume group
veggie2:
LV NAME             TYPE       LPs   PPs  PVs  LV STATE      MOUNT POINT
hd2                 jfs        103   103  1    open/syncd    /usr
hd1                 jfs        1     1    1    open/syncd    /home
hd3                 jfs        3     3    1    open/syncd    /tmp
hd9var              jfs        1     1    1    open/syncd    /var
file systems
Filesystem    Total KB    free %used   iused %iused Mounted on
/dev/hd4          8192     420   94%     909    44% /
/dev/hd9var       4096    2972   27%      87     8% /var
/dev/hd2        421888   10964   97%   17435    16% /usr
/dev/hd3         12288   11588    5%      49     1% /tmp
/dev/hd1          4096    3896    4%      26     2% /home
```

3. With the information obtained, restore each client machine.

Recovering When Using Different Hardware at the Recovery Site

You may have to recover your system using hardware that is different from that used when you backed up your database and created disaster recovery plan file. Before restoring the database, update the device configuration file included in the recovery plan file. After restoring the database, update the device configuration on the database.

This section describes a number of such situations in detail. If the hardware environment is different at the recovery site, you must update the device configuration file. TSM uses the device configuration file to access the devices that are needed to read the database backup volumes. The RECOVERY.VOLUMES.REQUIRED stanza in the plan file identifies the volumes that are needed to restore the database.

Automated SCSI Library at the Original Site and a Manual SCSI Library at the Recovery Site

Ensure that the DEFINE DRIVE and DEFINE LIBRARY commands in the device configuration file are valid for the new hardware configuration. For example, if an automated tape library was used originally and cannot be used at the recovery site, update the device configuration file. Include the DEFINE LIBRARY and DEFINE DRIVE commands that are needed to define the manual drive to be used. In this case, you must manually mount the backup volumes.

Note: If you are using an automated library, you may also need to update the device configuration file to specify the location of the database backup volume.

Here is an example of an original device configuration file, which describes an automated tape library:

```
/*  Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
  mountlimit=2 mountwait=60 mountretention=60
  prefix=tsm library=auto8mmlib

define library auto8mmlib libtype=scsi

define drive auto8mmlib 8mm_tape0 element=82 online=yes

define drive auto8mmlib 8mm_tape1 element=83 online=yes

define path server1 auto8mmlib srctype=server desttype=library
 device=/dev/lb4

define path server1 8mm_tape0 srctype=server desttype=drive
library=auto8mmlib device=/dev/mt1

define path server1 8mm_tape1 srctype=server desttype=drive
library=auto8mmlib device=/dev/mt2

/* LIBRARYINVENTORY SCSI AUTO8MMLIB KEV004 1 101*/
/* LIBRARYINVENTORY SCSI AUTO8MMLIB KEV005 3 101*/
```

Here is an example of the updated device configuration file when a manual library is used at the recovery site:

```
/* Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
```

```
  mountlimit=1 mountwait=60 mountretention=60 prefix=tsm
  library=manual8mm

define library manual8mm libtype=manual

define drive manual8mm 8mm_tape0

define path server1 8mm_tape0 srctype=server desttype=drive
library=manual8mm device=/dev/mt1
```

The following changes were made:

- In the device class definition, the library name was changed from AUTO8MMLIB to MANUAL8MM. The device class name remains the same because it is associated with the database backup volumes in the volume history file.
- The manual library, MANUAL8MM, was defined.
- A new drive, 8MM_TAPE0, was defined for the manual library.
- The comments that named the location of volumes in the automated library were removed.

After you restore the database, modify the device configuration file in the database. After starting the server, define, update, and delete your library and drive definitions to match your new configuration.

Note: If you are using an automated library, you may need to use the AUDIT LIBRARY command to update the server inventory of the library volumes.

Automated SCSI Library at the Original and Recovery Sites

Manually place the database backup volumes in the automated library and note the element numbers where you place them. Then update the comments in the device configuration file to identify the locations of those volumes.

Note: You may also need to audit the library after the database is restored in order to update the server inventory of the library volumes.

Here is an example of an original device configuration file, which describes an automated tape library:

```
/*  Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
  mountlimit=2 mountwait=60 mountretention=60
  prefix=tsm library=auto8mmlib

define library auto8mmlib libtype=scsi

define drive auto8mmlib 8mm_tape0 element=82 online=yes

define drive auto8mmlib 8mm_tape1 element=83 online=yes

define path server1 auto8mmlib srctype=server desttype=library
 device=/dev/lb4

define path server1 8mm_tape0 srctype=server desttype=drive
library=auto8mmlib device=/dev/mt1

define path server1 8mm_tape1 srctype=server desttype=drive
library=auto8mmlib device=/dev/mt2
```

```
/* LIBRARYINVENTORY SCSI AUTO8MMLIB KEV004 1 101*/
/* LIBRARYINVENTORY SCSI AUTO8MMLIB KEV005 3 101*/
```

Here is an example of the updated device configuration file when an automated library is used at the recovery site to read a database volume DBBK01:

```
/*  Device Configuration */

define devclass auto8mm_class devtype=8mm format=drive
  mountlimit=2 mountwait=60 mountretention=60
  prefix=tsm library=auto8mmlib

define library auto8mmlib libtype=scsi

define drive auto8mmlib 8mm_tape0 element=82 online=yes

define drive auto8mmlib 8mm_tape1 element=83 online=yes

define path server1 auto8mmlib srctype=server desttype=library
 device=/dev/lb4

define path server1 8mm_tape0 srctype=server desttype=drive
library=auto8mmlib device=/dev/mt1

define path server1 8mm_tape1 srctype=server desttype=drive
library=auto8mmlib device=/dev/mt2

/* LIBRARYINVENTORY SCSI AUTO8MMLIB DBBK01 1 101*/
```

In this example, database backup volume DBBK01 was placed in element 1 of the automated library. Then a comment is added to the device configuration file to identify the location of the volume. Tivoli Storage Manager needs this informatiion to restore the database restore. Comments that no longer apply at the recovery site are removed.

Managing Copy Storage Pool Volumes at the Recovery Site

The RECOVERY.VOLUMES.REQUIRED stanza in the recovery plan file identifies the required copy storage pool volumes. The restored server uses copy storage pool volumes to satisfy requests (for example, from backup/archive clients) and to restore primary storage pool volumes that were destroyed. These volumes must be available to the restored server. After the database is restored, you can handle copy storage pool volumes at the recovery site in three ways:

- Mount each volume as requested by TSM. If an automated library is used at the recovery site, check the volumes into the library.
- Check the volumes into an automated library before TSM requests them.
- Manually place the volumes in an automated library and audit the library to update the server inventory.

Note: If you are using an automated library, you may also need to audit the library after the database is restored in order to update the TSM inventory of the volumes in the library.

Tivoli Disaster Recovery Manager Checklist

The following checklist can help you set up Tivoli Disaster Recovery Manager.

Table 34. Tivoli Disaster Recovery Manager Checklist

Activity	Start Date	End Date	Status	Person Resp.	Backup Person
Plan for DRM					
Evaluate your disaster recovery requirements • What are the business priorities for recovering your clients? • Where is the recovery site? • Is the recovery site hot, warm, or cold? • Do the clients have connectivity to recovery server? • Who are the system and TSM administrators? • Will you need to return to the original site? • Where are the offsite backups stored? • How does the vault handle the backup media? • How are the backups packaged or processed? • Who provides the courier service?					
Evaluate the current storage pool backup implementation • What primary storage pools are being backed up? • When are the backups performed? • Will the backups remain onsite or be sent offsite? • Naming conventions for replacement volumes for primary storage pools					
Evaluate the current database backup implementation • When are the backups performed? • Backup purpose: offsite or onsite • Will you use snapshot database backups or full plus incremental database backups? • How long do you want to keep backup series? Verify that the values for copy storage pool REUSEDELAY and DRMDBBACKUPEXPIREDAYS are the same.					
Determine which primary storage pools are to be managed by DRM					
Determine which copy storage pools are to be managed by DRM • Offsite copy storage pools					
Where to Save the Recovery Plan File **Locally:** • What is the recovery plan file pathname prefix? • How will recovery plan files be made available at the recovery site? – Print and store offsite – Tape/diskette copy stored offsite – Copy sent/NFS to recovery site **On Another Server:** • What server is to be used as the target server? • What is the name of the target server's device class? • How long do you want to keep recovery plan files?					
Determine where you want to create the user-specified recovery instructions What is the prefix of the instructions pathname?					

Table 34. Tivoli Disaster Recovery Manager Checklist (continued)

Activity	Start Date	End Date	Status	Person Resp.	Backup Person
Analyze the sequence of steps related to the PREPARE command backup movement Document the flow of activities and timings • Sending of volumes offsite • Return of empty volumes • PREPARE timing					
Installation					
Receive and Install the TSM code					
License DRM • REGISTER LICENSE or • Update the server options					
Set DRM defaults Issue: • SET DRMDBBACKUPEXPIREDAYS to define the database backup expiration • SET DRMPRIMSTGPOOL to specify the DRM-managed primary storage pools • SET DRMCOPYSTGPOOL to specify the DRM-managed copy storage pools • SET DRMPLANVPOSTFIX to specify a character to be appended to new storage pools • SET DRMPLANPREFIX to specify the RPF prefix • SET DRMINSTRPREFIX to specify the user instruction file prefix • SET DRMNOTMOUNTABLENAME to specify the default location for media to be sent offsite • SET DRMCOURIERNAME to specify the default courier • SET DRMVAULTNAME to specify the default vault • SET DRMCMDFILENAME to specify the default file name to contain the commands specified with the CMD parameter on MOVE and QUERY DRMEDIA • SET DRMCHECKLABEL to specify whether volume labels are verified when checked out by the MOVE DRMEDIA command • SET DRMRPFEXPIREDAYS to specify a value for the frequency of RPF expiration (when plan files are stored on another server)					

Table 34. Tivoli Disaster Recovery Manager Checklist (continued)

Activity	Start Date	End Date	Status	Person Resp.	Backup Person
Define the site-specific recovery instructions Identify: • Target disaster recovery server location • Target server software requirements • Target server hardware requirements (storage devices) • TSM administrator contact • Courier name and telephone number • Vault location and contact person Create: • Enter the site-specific recovery instructions data into files created in the same path/HLQ as specified by SET DRMINSTRPREFIX					
Test Tivoli Disaster Recovery Manager					
Test the installation and customization • QUERY DRMSTATUS to display the DRM setup • Back up the primary storage pools • Back up the TSM database • QUERY DRMEDIA to list the backup volumes • MOVE DRMEDIA to move offsite • PREPARE to create the recovery plan file					
Examine the recovery plan file created					
Test the recovery plan file break out • awk script planexpl.awk • Locally written procedure					
Set up the schedules for automated functions					

The Disaster Recovery Plan File

The disaster recovery plan file contains the information required to recover a TSM server to the point in time represented by the last database backup operation that is completed before the plan is created. The plan is organized into stanzas, which you can break out into multiple files.

Breaking Out a Disaster Recovery Plan File

You can use an awk script or an editor to break out the stanzas into individual files. A sample procedure, *planexpl.awk.smp*, is shipped with DRM and is located in */usr/tivoli/tsm/server/bin* or wherever the server resides. You can modify this procedure for your installation. Store a copy of the procedure offsite for recovery.

Structure of the Disaster Recovery Plan File

The disaster recovery plan is divided into the following types of stanzas:

Command stanzas
: Consist of scripts (for example, batch programs or batch files) and TSM macros. You can view, print, and update these stanzas, and run them during recovery.

Note: The RECOVERY.SCRIPT.NORMAL.MODE and RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE stanzas contain the commands that invoke the scripts and macros contained in the other stanzas.

Instruction stanzas
Consist of recovery instructions specific to your site. You can view, print, and update these stanzas, and use them during recovery.

Server requirements stanzas
Include the database and recovery log requirements, device and volume requirements, and license information. You can view and print these stanzas, and use them during recovery.

Configuration file stanzas
Consist of the volume history, device configuration, and server options files.

Machine and recovery media stanzas
Consist of machine recovery instructions and information about machine hardware, software, and recovery media. You can print and update these stanzas, and use them during server recovery.

Table 35 lists the recovery plan file stanzas, and indicates what type of administrative processing is required during set up, routine operations, and disaster recovery. The table also indicates whether the stanza contains a macro, a script, or a configuration file.

Note: For tasks identified as **During setup or periodic updates**, DRM automatically collects this information for the plan.

Table 35. Administrative Tasks Associated with the Disaster Recovery Plan File

Stanza Name	Tasks
PLANFILE.DESCRIPTION	—
PLANFILE.TABLE.OF.CONTENTS	—
SERVER.REQUIREMENTS	—
RECOVERY.INSTRUCTIONS.GENERAL	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.INSTRUCTIONS.OFFSITE	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.INSTRUCTIONS.INSTALL	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.INSTRUCTIONS.DATABASE	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.INSTRUCTIONS.STGPOOL	**During setup or periodic updates:** Edit the source file associated with the stanza (optional)
RECOVERY.VOLUMES.REQUIRED	**During routine processing:** MOVE DRMEDIA
RECOVERY.DEVICES.REQUIRED	—

Table 35. Administrative Tasks Associated with the Disaster Recovery Plan File (continued)

Stanza Name	Tasks
RECOVERY.SCRIPT. DISASTER.RECOVERY.MODE script	**During disaster recovery:** Edit and run (optional)
RECOVERY.SCRIPT. NORMAL.MODE script	**During disaster recovery:** Edit and run (optional)
LOGANDDB.VOLUMES.CREATE script	**During disaster recovery:** Edit and run (optional)
LOG.VOLUMES	**During disaster recovery:** Optionally edit/copy
DB.VOLUMES	**During disaster recovery:** Optionally edit/copy
LOGANDDB.VOLUMES.INSTALL script	**During disaster recovery:** Edit and run (optional)
LICENSE.REGISTRATION macro	**During disaster recovery:** Edit and run (optional)
COPYSTGPOOL.VOLUMES.AVAILABLE macro	**During routine processing:** MOVE DRMEDIA **During disaster recovery:** Edit and run (optional)
COPYSTGPOOL.VOLUMES.DESTROYED macro	**During routine processing:** MOVE DRMEDIA **During disaster recovery:** Edit and run (optional)
PRIMARY.VOLUMES.DESTROYED macro	**During disaster recovery:** Edit and run (optional)
PRIMARY.VOLUMES.REPLACEMENT.CREATE script	**During disaster recovery:** Edit and run (optional)
PRIMARY.VOLUMES.REPLACEMENT macro	**During disaster recovery:** Edit and run (optional)
STGPOOLS.RESTORE macro	**During disaster recovery:** Edit and run (optional)
VOLUME.HISTORY.FILE configuration file	**During disaster recovery:** Copy (optional)
DEVICE.CONFIGURATION.FILE configuration file	**During disaster recovery:** Edit and copy (optional)
DSMSERV.OPT.FILE configuration file	**During disaster recovery:** Edit and copy (optional)
LICENSE.INFORMATION	—
MACHINE.GENERAL.INFORMATION	**During setup or periodic updates:** Issue DEFINE MACHINE ADSMSERVER=YES (optional)
MACHINE.RECOVERY.INSTRUCTIONS	**During setup or periodic updates:** Issue INSERT MACHINE RECOVERYINSTRUCTIONS (optional)
MACHINE.RECOVERY.CHARACTERISTICS	**During setup or periodic updates:** Issue INSERT MACHINE CHARACTERISTICS (optional)

Table 35. Administrative Tasks Associated with the Disaster Recovery Plan File (continued)

Stanza Name	Tasks
MACHINE.RECOVERY.MEDIA	**During setup or periodic updates:** Issue DEFINE RECOVERYMEDIA and DEFINE RECMEDMACHASSOCIATION (optional)

Example Disaster Recovery Plan File

This section contains an example of a disaster recovery plan file and information about each stanza. The disaster recovery plan file has been divided into separate figures that correlate to the descriptions of specific stanzas within each figure.

Description and Table of Contents Stanzas

PLANFILE.DESCRIPTION

Identifies the server for this recovery plan, and the date and time the plan is created.

```
begin PLANFILE.DESCRIPTION

Recovery Plan for Server DESIGN_DEPARTMENT
Created by DRM PREPARE on 02/11/2000 10:20:34
Server for AIX-RS/6000 - Version 4, Release 1, Level x.x/x.x

end PLANFILE.DESCRIPTION
```

Figure 87. Description Stanza

PLANFILE.TABLE.OF.CONTENTS

Lists the stanzas documented in this plan.

```
 begin PLANFILE.TABLE.OF.CONTENTS

PLANFILE.DESCRIPTION
PLANFILE.TABLE.OF.CONTENTS

Server Recovery Stanzas:
  SERVER.REQUIREMENTS
  RECOVERY.INSTRUCTIONS.GENERAL
  RECOVERY.INSTRUCTIONS.OFFSITE
  RECOVERY.INSTRUCTIONS.INSTALL
  RECOVERY.VOLUMES.REQUIRED
  RECOVERY.DEVICES.REQUIRED
  RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script
  RECOVERY.SCRIPT.NORMAL.MODE script
  LOGANDDB.VOLUMES.CREATE script
  LOG.VOLUMES
  DB.VOLUMES
  LOGANDDB.VOLUMES.INSTALL script
  LICENSE.REGISTRATION macro
  COPYSTGPOOL.VOLUMES.AVAILABLE macro
  COPYSTGPOOL.VOLUMES.DESTROYED macro
  PRIMARY.VOLUMES.DESTROYED macro
  PRIMARY.VOLUMES.REPLACEMENT.CREATE script
  PRIMARY.VOLUMES.REPLACEMENT macro
  STGPOOLS.RESTORE macro
  VOLUME.HISTORY.FILE
  DEVICE.CONFIGURATION.FILE
  DSMSERV.OPT.FILE

Machine Description Stanzas:
  MACHINE.GENERAL.INFORMATION
  MACHINE.RECOVERY.INSTRUCTIONS
  MACHINE.CHARACTERISTICS
  MACHINE.RECOVERY.MEDIA.REQUIRED

end PLANFILE.TABLE.OF.CONTENTS
```

Figure 88. Table of Contents Stanza

Server Requirements Stanza

SERVER.REQUIREMENTS

Identifies the database and recovery log storage requirements for the server. The replacement server must have enough disk space to install the database and recovery log volumes. This stanza also identifies the directory where the server executable resided when the server was started. If the server executable is in a different directory on the replacement server, edit the plan file to account for this change.

If you use links to the server executable file, you must create the links on the replacement machine or modify the following plan file stanzas:

- RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE
- LOGANDDB.VOLUMES.CREATE
- LOGANDDB.VOLUMES.INSTALL
- PRIMARY.VOLUMES.REPLACEMENT.CREATE

```
 begin SERVER.REQUIREMENTS

Database Requirements Summary:

    Available Space (MB): 20
  Assigned Capacity (MB): 20
        Pct. Utilization: 2.2
Maximum Pct. Utilization: 2.2
        Physical Volumes: 2

Recovery Log Requirements Summary:

    Available Space (MB): 20
  Assigned Capacity (MB): 20
        Pct. Utilization: 4.4
Maximum Pct. Utilization: 4.8
        Physical Volumes: 2
Server Executable Location: /usr/tivoli/tsm/server/bin
end SERVER.REQUIREMENTS
```

Figure 89. Server Requirements Stanza

Recovery Instructions Stanzas

The administrator enters recovery instructions into source files that the PREPARE command includes in the plan files. See "Specifying Recovery Instructions for Your Site" on page 550 for details.

Note: In the following descriptions, *prefix* represents the prefix portion of the file name. See "Specifying Defaults for the Disaster Recovery Plan File" on page 546 for details.

RECOVERY.INSTRUCTIONS.GENERAL

Identifies site-specific instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.GENERAL. The instructions should include the recovery strategy, key contact names, an overview of key applications backed up by this server, and other relevant recovery instructions.

```
begin RECOVERY.INSTRUCTIONS.GENERAL

 This server contains the backup and archive data for FileRight Company
 accounts receivable system. It also is used by various end users in the
 finance and materials distribution organizations.
 The storage administrator in charge of this server is Jane Doe 004-001-0006.
 If a disaster is declared, here is the outline of steps that must be completed.
 1. Determine the recovery site. Our alternate recovery site vendor is IBM
    BRS in Tampa, Fl, USA 213-000-0007.
 2. Get the list of required recovery volumes from this recovery plan file
    and contact our offsite vault so that they can start pulling the
    volumes for transfer to the recovery site.
 3. etc...

end RECOVERY.INSTRUCTIONS.GENERAL
```

Figure 90. Recovery Instructions General Stanza

RECOVERY.INSTRUCTIONS.OFFSITE

Contains instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.OFFSITE. The instructions should include the name and location of the offsite vault, and how to contact the vault (for example, a name and phone number).

```
begin RECOVERY.INSTRUCTIONS.OFFSITE

 Our offsite vaulting vendor is OffsiteVault Inc.
 Their telephone number is 514-555-2341. Our account rep is Joe Smith.
 Our account number is 1239992. Their address is ...
 Here is a map to their warehouse ...
 Our courier is ...

end RECOVERY.INSTRUCTIONS.OFFSITE
```

Figure 91. Recovery Instructions Offsite Stanza

RECOVERY.INSTRUCTIONS.INSTALL

Contains instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.INSTALL. The instructions should include how to rebuild the base server machine and the location of the system image backup copies.

```
begin RECOVERY.INSTRUCTIONS.INSTALL

 The base server system is AIX 4.3 running on an RS6K model 320.
 Use mksysb volume serial number svrbas to restore this system image.
 A copy of this mksysb tape is stored at the vault. There is also a copy
 in bldg 24 room 4 cabinet a. The image includes the server code.
 The system programmer responsible for this image is Fred Myers.
 Following are the instructions to do a mksysb based OS install:

end RECOVERY.INSTRUCTIONS.INSTALL
```

Figure 92. Recovery Instructions Install Stanza

RECOVERY.INSTRUCTIONS.DATABASE

Contains instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.DATABASE. The instructions should include how to prepare for the database recovery. For example, you may enter instructions on how to initialize or load the backup volumes for an automated library. No sample of this stanza is provided.

RECOVERY.INSTRUCTIONS.STGPOOL

Contains instructions that the administrator has entered in the file identified by *prefix* RECOVERY.INSTRUCTIONS.STGPOOL. The instructions should include the names of your software applications and the copy storage pool names containing the backup of these applications. No sample of this stanza is provided.

Volume and Device Requirements Stanzas

RECOVERY.VOLUMES.REQUIRED

Provides a list of the database backup and copy storage pool volumes required to recover the server. This list can include both virtual volumes and nonvirtual volumes. A database backup volume is included if it is part of the most recent database backup series. A copy storage pool volume is included if it is not empty and not marked *destroyed*.

If you are using a nonvirtual volume environment and issuing the MOVE DRMEDIA command, a blank location field means that the volumes are onsite and available to the server. This volume list can be used in periodic audits of the volume inventory of the courier and vault. You can use the list to collect the required volumes before recovering the server.

For virtual volumes, the location field contains the target server name.

```
 begin RECOVERY.VOLUMES.REQUIRED

Volumes required for data base restore
  Location = OffsiteVault Inc.
  Device Class = LIB8MM
  Volume Name =
   TPBK08
 Location = OffsiteVault Inc.
  Device Class = LIB8MM
  Volume Name =
   TPBK06

Volumes required for storage pool restore
 Location = OffsiteVault Inc.
  Copy Storage Pool = CSTORAGEPF
  Device Class = LIB8MM
  Volume Name =
   TPBK05
   TPBK07

end RECOVERY.VOLUMES.REQUIRED
```

Figure 93. Volume Requirements Stanza

RECOVERY.DEVICES.REQUIRED

Provides details about the devices needed to read the backup volumes.

```
begin RECOVERY.DEVICES.REQUIRED

 Purpose: Description of the devices required to read the
          volumes listed in the recovery volumes required stanza.

             Device Class Name: LIB8MM
        Device Access Strategy: Sequential
            Storage Pool Count: 2
                   Device Type: 8MM
                        Format: DRIVE
         Est/Max Capacity (MB): 4.0
                   Mount Limit: 2
              Mount Wait (min): 60
         Mount Retention (min): 10
                  Label Prefix: TIVSM
                       Library: RLLIB
                     Directory:
Last Update by (administrator): Bill
         Last Update Date/Time: 12/11/2000 10:18:34

end RECOVERY.DEVICES.REQUIRED
```

Figure 94. Volume and Device Requirements Stanzas

Disaster Recovery Mode Stanza

RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE

Contains a script with the commands needed to recover the server. You can use the script as a guide and run the commands from a command line. Or you can copy it to a file, modify it and the files it refers to, and run the script. You may need to modify the script because of differences between the original and the replacement systems. At the completion of these steps, client requests for file restores are satisfied directly from copy storage pool volumes.

The disaster recovery plan issues commands using the administrative client. The disaster recovery plan file issues commands using the administrative client. Ensure that the path to the administrative client is established before running the script. For example, set the shell variable PATH or update the scripts with the path specification for the administrative client.

The commands in the script do the following:

- Restore the server options, volume history, and device configuration information files.
- Invoke the scripts contained in the LOGANDDB.VOLUMES.CREATE and LOGANDDB.VOLUMES.INSTALL stanzas.

  Attention: When this script runs, any log volumes or database volumes with the same names as those named in the plan are *removed* (see LOGANDDB.VOLUMES.CREATE under “Create and Install Database and Recovery Log Volumes Stanzas” on page 588). In most disaster recoveries, the TSM server is installed on a new machine. When this script is run, it is assumed that there is no TSM data in the log or database volumes. TSM installation includes the creation of database and recovery log volumes. If you have created a log volume or a database volume (for example, for testing), and you want to preserve the contents, you must take some action such as renaming the volume or copying the contents before executing this script.
- Invoke the macros contained in the following stanzas:

  - LICENSE.REGISTRATION
  - COPYSTGPOOL.VOLUMES.AVAILABLE
  - COPYSTGPOOL.VOLUMES.DESTROYED
  - PRIMARY.VOLUMES.DESTROYED.

To help understand the operations being performed in this script, see “Backup and Recovery Scenarios” on page 539.

To invoke this script, specify the following positional parameters:

- $1 (the administrator ID)
- $2 (the administrator password)
- $3 (the server ID as specified in the dsm.sys file)

  Note: The default location for dsm.sys is */usr/tivoli/tsm/client/admin/bin*.

For example, to invoke this script using an administrator ID of *don*, password of *mox*, server name of *prodtsm*, enter the following command:

```
planprefix/RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE don mox prodtsm
```

For more information, see the entry for the recovery plan prefix in Table 33 on page 547.

```
 begin RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script

#!/bin/ksh
set -x

 # Purpose: This script contains the steps required to recover the server
 #   to the point where client restore requests can be satisfied
 #   directly from available copy storage pool volumes.
 # Note: This script assumes that all volumes necessary for the restore have
 #   been retrieved from the vault and are available. This script assumes
 #   the recovery  environment is compatible (essentially the same) as the
 #   original.  Any deviations require modification to this script and the
 #   macros and shell scripts it runs.  Alternatively, you can use this
 #   script as a guide, and manually execute each step.

if [ -z "$1" -o -z "$2" -o -z "$3" ]
then
  print "Specify the following positional parameters:"
  print "administrative client ID, password, and server ID."
  print "Script stopped."
  exit
fi
 # Set the  server working directory
cd /usr/tivoli/tsm/server/bin

 # Restore server options, volume history, device configuration files.
cp /prepare/DSMSERV.OPT.FILE \
    /usr/tivoli/tsm/server/bindsmserv.optx
cp /prepare/VOLUME.HISTORY.FILE \
    /usr/tivoli/tsm/server/binvolhistory.txtx
cp /prepare/DEVICE.CONFIGURATION.FILE \
    /usr/tivoli/tsm/server/bindevconfig.txtx

export DSMSERV_CONFIG=/usr/tivoli/tsm/server/bindsmserv.optx

export DSMSERV_DIR=/opt/adsmserv/bin
```

Figure 95. Disaster Recovery Mode Script (Part 1 of 2)

```
 # Create and format log and database files.
/prepare/LOGANDDB.VOLUMES.CREATE 2>&1 \
| tee /prepare/LOGANDDB.VOLUMES.CREATE.log

 # Initialize the log and database files.
/prepare/LOGANDDB.VOLUMES.INSTALL 2>&1 \
| tee /prepare/LOGANDDB.VOLUMES.INSTALL.log

 # Restore the  server database to latest version backed up per the
 # volume history file.
/usr/tivoli/tsm/server/bindsmserv restore db todate=08/11/2000 totime=10:20:22

 # Start the server.
nohup /usr/tivoli/tsm/server/bindsmserv &
print Please start new  server console with command dsmadmc -CONSOLE.
print Press enter to continue recovery script execution.
read pause

 # Register  Server Licenses.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT
  -OUTFILE=/prepare/LICENSE.REGISTRATION.log
    macro /prepare/LICENSE.REGISTRATION.mac

 # Tell  Server these copy storage pool volumes are available for use.
 # Recovery Administrator: Remove from macro any volumes not obtained from vault.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/COPYSTGPOOL.VOLUMES.AVAILABLE.log \
     macro /prepare/COPYSTGPOOL.VOLUMES.AVAILABLE

 # Volumes in this macro were not marked as 'offsite' at the time
 # PREPARE ran. They were likely destroyed in the disaster.
 # Recovery Administrator: Remove from macro any volumes not destroyed.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/COPYSTGPOOL.VOLUMES.DESTROYED.log \
     macro /prepare/COPYSTGPOOL.VOLUMES.DESTROYED
 # Mark primary storage pool volumes as ACCESS=DESTROYED.
 # Recovery administrator: Remove from macro any volumes not destroyed.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/PRIMARY.VOLUMES.DESTROYED.log \
     macro /prepare/PRIMARY.VOLUMES.DESTROYED

end RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script
```

Figure 95. Disaster Recovery Mode Script (Part 2 of 2)

Normal Mode Stanza

RECOVERY.SCRIPT.NORMAL.MODE

Contains a script with the commands needed to restore the server primary storage pools. You can use the script as a guide and run the commands from a command line. Or you can copy it to a file, modify it and the files it refers to, and run the script. You may need to modify the script because of differences between the original and the replacement systems.

The disaster recovery plan issues commands using the administrative client. The disaster recovery plan file issues commands using the administrative client. Ensure that the path to the administrative client is established before running the script. For example, set the shell variable PATH or update the scripts with the path specification for the administrative client.

At the completion of these steps, client requests for file restores are satisfied from primary storage pool volumes. Clients should also be able to resume file backup, archive, and migration functions.

This script invokes the script contained in the PRIMARY.VOLUMES.REPLACEMENT.CREATE stanza: It also invokes the macros contained in the following stanzas:
PRIMARY.VOLUMES.REPLACEMENT
STGPOOLS.RESTORE

To help understand the operations being performed in this script, see “Backup and Recovery Scenarios” on page 539.

To invoke this script, the following positional parameters must be specified:

- $1 (the administrator ID)
- $2 (the administrator password)
- $3 (the server ID as specified in the dsm.sys file)

For example, to invoke this script using an administrator ID of *don*, password of *mox*, server name of *prodtsm*, enter the following command:

```
planprefix/RECOVERY.SCRIPT.NORMAL.MODE don mox prodtsm
```

For more information, see the entry for the recovery plan prefix in Table 33 on page 547.

```
begin RECOVERY.SCRIPT.NORMAL.MODE script
#!/bin/ksh
set -x

 # Purpose: This script contains the steps required to recover the server
 #          primary storage pools. This mode allows you to return the
 #          copy storage pool volumes to the vault and to run the
 #          server as normal.
 # Note: This script assumes that all volumes necessary for the restore
 #   have been retrieved from the vault and are available. This script
 #   assumes the recovery  environment is compatible (essentially the
 #   same) as the original. Any deviations require modification to this
 #   script and the macros and shell scripts it runs. Alternatively,
 #   you can use this script as a guide, and manually execute each step.

if [ -z "$1" -o -z "$2" -o -z "$3" ]
then
  print "Specify the following positional parameters:"
  print "administrative client ID, password, and server ID."
  print "Script stopped."
  exit
fi

 # Create replacement volumes in the primary storage pools (If any
 # are implemented as disk but not logical volume.)
 # Recovery administrator: Edit script for your replacement volumes.
/prepare/PRIMARY.VOLUMES.REPLACEMENT.CREATE 2>&1 \
| tee /prepare/PRIMARY.VOLUMES.REPLACEMENT.CREATE.log

 # Define replacement volumes in the primary storage pools. Must
 # have different name than original.
 # Recovery administrator: Edit macro for your replacement volumes.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/PRIMARY.VOLUMES.REPLACEMENT.log \
     macro /prepare/PRIMARY.VOLUMES.REPLACEMENT

 # Restore the primary storage pools from the copy storage pools.
dsmadmc -id=$1 -pass=$2 -serv=$3 -ITEMCOMMIT \
  -OUTFILE=/prepare/STGPOOLS.RESTORE.log \
     macro /prepare/STGPOOLS.RESTORE

end RECOVERY.SCRIPT.NORMAL.MODE script
```

Figure 96. Normal Mode Script

Create and Install Database and Recovery Log Volumes Stanzas

LOGANDDB.VOLUMES.CREATE

Contains a script with the commands needed to recreate the database and log volumes. You can use the script as a guide and issue the commands as needed from a command line, or you can copy it to a file, modify it, and run it. This script is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

The plan assumes that the volume formatting command (DSMFMT) resides in the same directory as the server executable indicated in the stanza SERVER.REQUIREMENTS.

```
begin LOGANDDB.VOLUMES.CREATE script
#!/bin/ksh
set -x
 # Purpose:  Create log and database volumes.
 # Recovery Administrator: Run this to format  server log and
 #  database volumes.
  print Remove database volume /usr/tivoli/tsm/server/bindb01x.
 rm -f /usr/tivoli/tsm/server/bindb01x

  print Create  database volume /usr/tivoli/tsm/server/bindb01x 12M
 /usr/tivoli/tsm/server/bindsmfmt -m -db /usr/tivoli/tsm/server/bindb01x 12M

  print Remove database volume /usr/tivoli/tsm/server/bindb02x.
 rm -f /usr/tivoli/tsm/server/bindb02x

  print Create  database volume /usr/tivoli/tsm/server/bindb02x 8M
 /usr/tivoli/tsm/server/bindsmfmt -m -db /usr/tivoli/tsm/server/bindb02x 8

  print Remove log volume /usr/tivoli/tsm/server/binlg01x.
 rm -f /usr/tivoli/tsm/server/binlg01x

  print Create  log volume /usr/tivoli/tsm/server/binlg01x 12M
 /usr/tivoli/tsm/server/bindsmfmt -m -log /usr/tivoli/tsm/server/binlg01x 12M

  print Remove log volume /usr/tivoli/tsm/server/binlg02x.
 rm -f /usr/tivoli/tsm/server/bin..lg02x

  print Create  log volume /usr/tivoli/tsm/server/binlg02x 8M
 /usr/tivoli/tsm/server/bindsmfmt -m -log /usr/tivoli/tsm/server/binlg02x 8

end LOGANDDB.VOLUMES.CREATE script
```

Figure 97. Create Database and Recovery Log Volumes Stanza

LOG.VOLUMES

Contains the names of the log volumes to be initialized. The contents of this stanza must be placed into a separate file to be used by the LOGANDDB.VOLUMES.INSTALL script.

```
begin LOG.VOLUMES
/usr/tivoli/tsm/server/binlg01x
/usr/tivoli/tsm/server/binlg02x

end LOG.VOLUMES
```

Figure 98. Recovery Log Volumes Stanza

DB.VOLUMES

Contains the names of the database volumes to be initialized. The contents of this stanza must be placed into a separate file to be used by the LOGANDDB.VOLUMES.INSTALL script.

```
begin DB.VOLUMES

/usr/tivoli/tsm/server/bindb01x
/usr/tivoli/tsm/server/bindb02x

end DB.VOLUMES
```

Figure 99. Database Volume Stanza

LOGANDDB.VOLUMES.INSTALL

Contains a script with the commands required to initialize the database and log volumes. This script is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

```
 begin LOGANDDB.VOLUMES.INSTALL script

#!/bin/ksh
set -x

 # Purpose: Initialize the log and database volumes.
 # Recovery Administrator: Run this to initialize an  server.

 /usr/tivoli/tsm/server/bindsmserv install \
   2 FILE:/prepare/LOG.VOLUMES \
   2 FILE:/prepare/DB.VOLUMES

end LOGANDDB.VOLUMES.INSTALL script
```

Figure 100. Install Database and Recovery Log Volumes Stanza

License Registration Stanza

LICENSE.REGISTRATION

Contains a macro to register your server licenses. This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

```
begin LICENSE.REGISTRATION macro

 /* Purpose: Register the  Server licenses by specifying the names      */
 /*  of the enrollment certificate files necessary to recreate the      */
 /*  licenses that existed in the  server.                              */
 /* Recovery Administrator: Review licenses and add or delete licenses  */
 /*  as necessary.                                                      */

register license file(50client.lic)
register license file(network.lic)
register license file(drm.lic)

end LICENSE.REGISTRATION macro
```

Figure 101. License Registration Macro Stanza

Copy Storage Pool Volumes Stanzas

COPYSTGPOOL.VOLUMES.AVAILABLE

Contains a macro to mark copy storage pool volumes that were moved offsite and then moved back onsite. This stanza does not include copy storage pool virtual volumes. You can use the information as a guide and issue the administrative commands, or you can copy it to a file, modify it, and run it. This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

After a disaster, compare the copy storage pool volumes listed in this stanza with the volumes that were moved back onsite. You should remove entries from this stanza for any missing volumes.

```
begin COPYSTGPOOL.VOLUMES.AVAILABLE macro

 /* Purpose: Mark copy storage pool volumes as available for use in recovery. */
 /* Recovery Administrator: Remove any volumes that have not been obtained    */
 /*   from the vault or are not available for any reason.                    */
 /* Note: It is possible to use the mass update capability of the            */
 /*   UPDATE command instead of issuing an update for each volume. However,   */
 /*   the 'update by volume' technique used here allows you to select        */
 /*   a subset of volumes to be processed.                                   */

 upd vol TPBK05 acc=READW wherestg=CSTORAGEPF
 upd vol TPBK07 acc=READW wherestg=CSTORAGEPF

end COPYSTGPOOL.VOLUMES.AVAILABLE macro
```

Figure 102. Copy Storage Pool Volumes Available Stanza

COPYSTGPOOL.VOLUMES.DESTROYED

Contains a macro to mark copy storage pool volumes as unavailable if the volumes were onsite at the time of the disaster. This stanza does not include copy storage pool virtual volumes. These volumes are considered offsite and have not been destroyed in a disaster. You can use the information as a guide and issue the administrative commands from a command line, or you can copy it to a file, modify it, and run it. This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

After a disaster, compare the copy storage pool volumes listed in this stanza with the volumes that were left onsite. If you have any of the volumes and they are usable, you should remove their entries from this stanza.

```
begin COPYSTGPOOL.VOLUMES.DESTROYED macro

 /* Purpose: Mark destroyed copy storage pool volumes as unavailable.      */
 /*   Volumes in this macro were not marked as 'offsite' at the time the   */
 /*   PREPARE ran. They were likely destroyed in the disaster.             */
 /* Recovery Administrator: Remove any volumes that were not destroyed.    */


end COPYSTGPOOL.VOLUMES.DESTROYED macro
```

Figure 103. Copy Storage Pool Volumes Destroyed Stanza

Primary Storage Volumes Stanzas

PRIMARY.VOLUMES.DESTROYED

Contains a macro to mark primary storage pool volumes as *destroyed* if the volumes were onsite at the time of disaster. You can use the information as a guide and run the administrative commands from a command line, or you can copy it to a file, modify it, and run it. This macro is invoked by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

During recovery, compare the primary storage pool volumes listed in this stanza with the volumes that were onsite. If you have any of the volumes and they are usable, remove their entries from the stanza.

This stanza does not include primary storage pool virtual volumes. These volumes are considered offsite and have not been destroyed in a disaster.

```
begin PRIMARY.VOLUMES.DESTROYED macro

 /* Purpose: Mark primary storage pool volumes as ACCESS=DESTROYED.       */
 /* Recovery administrator: Delete any volumes listed here                */
 /*   that you do not want to recover.                                    */
 /* Note: It is possible to use the mass update capability of the         */
 /*   UPDATE command instead of issuing an update for each volume. However*/
 /*   the 'update by volume' technique used here allows you to select     */
 /*   a subset of volumes to be marked as destroyed.                      */

 upd vol /usr/tivoli/tsm/server/binbk02 acc=DESTROYED wherestg=BACKUPPOOL
 upd vol /usr/tivoli/tsm/server/binbk01x acc=DESTROYED wherestg=BACKUPPOOL
 upd vol /usr/tivoli/tsm/server/binbk03 acc=DESTROYED wherestg= BACKUPPOOLF
 upd vol BACK4X acc=DESTROYED wherestg=BACKUPPOOLT

end PRIMARY.VOLUMES.DESTROYED macro
```

Figure 104. Primary Storage Volumes Destroyed Stanza

PRIMARY.VOLUMES.REPLACEMENT.CREATE

Contains a script with the commands needed to recreate the primary disk storage pool volumes. You can use the script as a guide and run the commands from a command line, or you can copy thew script to a file, modify it, and run it. This script is invoked by the RECOVERY.SCRIPT.NORMAL.MODE script.

The plan file assumes that the volume formatting program (DSMFMT) resides in the same directory as the server executable indicated in the stanza SERVER.REQUIREMENTS.

The SET DRMPLANVPOSTFIX command adds a character to the end of the names of the original volumes listed in this stanza. This character does the following:

- Improves retrievability of volume names that require renaming in the stanzas. Before using the volume names, change these names to new names that are valid for the device class and valid on the replacement system.
- Generates a new name that can be used by the replacement server. Your naming convention must take into account the appended character.

Notes:

1. Replacement primary volume names must be different from any other original volume name or replacement name.
2. The RESTORE STGPOOL command restores storage pools on a logical basis. There is no one-to-one relationship between an original volume and its replacement.

3. There will be entries for the same volumes in PRIMARY.VOLUMES.REPLACEMENT.

This stanza does not include primary storage pool virtual volumes, because these volumes are considered offsite and have not been destroyed in a disaster.

```
begin PRIMARY.VOLUMES.REPLACEMENT.CREATE script

#!/bin/ksh
set -x

 # Purpose: Create replacement volumes for primary storage pools that
 #   use device class DISK.
 # Recovery administrator: Edit this section for your replacement
 #   volume names. New name must be unique, i.e. different from any
 #   original or other new name.

   print Replace /usr/tivoli/tsm/server/binbk02 DISK 16M in BACKUPPOOL
 /usr/tivoli/tsm/server/bindsmfmt -m -data /usr/tivoli/tsm/server/binbk02@ 16

   print Replace /usr/tivoli/tsm/server/binbk01x DISK 5M in BACKUPPOOL
 /usr/tivoli/tsm/server/bindsmfmt -m -data /usr/tivoli/tsm/server/binbk01x@ 5

end PRIMARY.VOLUMES.REPLACEMENT.CREATE script
```

Figure 105. Primary Storage Volumes Replacement Stanza

PRIMARY.VOLUMES.REPLACEMENT

Contains a macro to define primary storage pool volumes to the server. You can use the macro as a guide and run the administrative commands from a command line, or you can copy it to a file, modify it, and execute it. This macro is invoked by the RECOVERY.SCRIPT.NORMAL.MODE script.

Primary storage pool volumes with entries in this stanza have at least one of the following three characteristics:

1. Original volume in a storage pool whose device class was DISK.
2. Original volume in a storage pool with MAXSCRATCH=0.
3. Original volume in a storage pool and volume scratch attribute=no.

The SET DRMPLANVPOSTFIX command adds a character to the end of the names of the original volumes listed in this stanza. This character does the following:

- Improves the retrievability of volume names that must be renamed in the stanzas. Before using the volume names, change these names to new names that are valid for the device class on the replacement system.
- Generates a new name that can be used by the replacement server. Your naming convention must take into account the appended character.

Notes:

1. Replacement primary volume names must be different from any other original volume name or replacement name.
2. The RESTORE STGPOOL command restores storage pools on a logical basis. There is no one-to-one relationship between an original volume and its replacement.

3. There could be entries for the same volume in PRIMARY.VOLUMES.REPLACEMENT.CREATE and PRIMARY.VOLUMES.REPLACEMENT if the volume has a device class of DISK.

This stanza does not include primary storage pool virtual volumes. These volumes are considered offsite and have not been destroyed in a disaster.

```
begin PRIMARY.VOLUMES.REPLACEMENT macro

 /* Purpose: Define replacement primary storage pool volumes for either:  */
 /*   1. Original volume in a storage pool whose device class was DISK.   */
 /*   2. Original volume in a storage pool with MAXSCRATCH=0.             */
 /*   3. Original volume in a storage pool and volume scratch=no.         */
 /* Recovery administrator: Edit this section for your replacement        */
 /*   volume names. New name must be unique, i.e. different from any      */
 /*   original or other new name.                                         */

   /* Replace /usr/tivoli/tsm/server/binbk02 DISK 16M in BACKUPPOOL */
 def vol BACKUPPOOL /usr/tivoli/tsm/server/binbk02@ acc=READW

   /* Replace /usr/tivoli/tsm/server/binbk01x DISK 5M in BACKUPPOOL */
 def vol BACKUPPOOL /usr/tivoli/tsm/server/binbk01x@ acc=READW

   /* Replace /usr/tivoli/tsm/server/binbk03 FILES 4M in BACKUPPOOLF */
 def vol BACKUPPOOLF /usr/tivoli/tsm/server/binbk03@ acc=READW

   /* Replace BACK4X COOL8MM 0M in BACKUPPOOLT */
 def vol BACKUPPOOLT BACK4X@ acc=READW

end PRIMARY.VOLUMES.REPLACEMENT macro
```

Figure 106. Primary Storage Volumes Replacement Stanza

Storage Pools Restore Stanza

STGPOOLS.RESTORE

Contains a macro to restore the primary storage pools. You can use it as a guide and execute the administrative commands from a command line. You can also can copy it to a file, modify it, and execute it. This macro is invoked by the RECOVERY.SCRIPT.NORMAL.MODE script.

This stanza does not include primary storage pool virtual volumes. These volumes are considered offsite and have not been destroyed in a disaster.

```
begin STGPOOLS.RESTORE macro

/* Purpose: Restore the primary storage pools from copy storage pool(s). */
/* Recovery Administrator: Delete entries for any primary storage pools  */
/*   that you do not want to restore.                                   */

 restore stgp ARCHIVEPOOL
 restore stgp BACKUPPOOL
 restore stgp BACKUPPOOLF
 restore stgp BACKUPPOOLT
 restore stgp SPACEMGPOOL

end STGPOOLS.RESTORE macro
```

Figure 107. Storage Pools Restore Stanza

Configuration Stanzas

VOLUME.HISTORY.FILE

Contains a copy of the volume history information when the recovery plan was created. The DSMSERV RESTORE DB command uses the volume history file to determine what volumes are needed to restore the database. It is used by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

The following rules determine where to place the volume history file at restore time:

- If the server option file contains VOLUMEHISTORY options, the server uses the fully qualified file name associated with the first entry. If the file name does not begin with a directory specification (for example, '.' or '/'), the server uses the prefix *volhprefix*.
- If the server option file does not contain VOLUMEHISTORY options, the server uses the default name *volhprefix* followed by *drmvolh.txt*. For example, if *volhprefix* is /usr/tivoli/tsm/server/bin, the file name is /usr/tivoli/tsm/server/bindrmvolh.txt.

Note: The *volhprefix* is set based on the following:

- If the environmental variable DSMSERV_DIR has been defined, it is used as the *volhprefix*.
- If the environmental variable DSMSERV_DIR has not been defined, the directory where the server is started from is used as the *volhprefix*.

If a fully qualified file name was not specified in the server options file for the VOLUMEHISTORY option, the server adds it to the DSMSERV.OPT.FILE stanza.

```
 begin VOLUME.HISTORY.FILE

*************************************************************************
*
*              Tivoli Storage Manager Sequential Volume Usage History
*                                    Updated 02/11/2000 10:20:34
*
*     Operation         Volume   Backup Backup Volume Device      Volume
*     Date/Time          Type    Series Oper.   Seq   Class Name  Name
*************************************************************************
 2000/08/11 10:18:43  STGNEW          0      0      0 COOL8MM     BACK4X
 2000/08/11 10:18:43  STGNEW          0      0      0 FILES       BK03
 2000/08/11 10:18:46  STGNEW          0      0      0 LIB8MM      TPBK05
* Location for volume TPBK06 is: 'Ironvault Inc.'
 2000/08/11 10:19:23  BACKUPFULL      1      0      1 LIB8MM      TPBK06
 2000/08/11 10:20:03  STGNEW          0      0      0 LIB8MM      TPBK07
 2000/08/11 10:20:22  BACKUPINCR      1      1      1 LIB8MM      TPBK08

end VOLUME.HISTORY.FILE
```

Figure 108. Volume History File Stanza

DEVICE.CONFIGURATION.FILE

Contains a copy of the server device configuration information when the recovery plan was created. The DSMSERV RESTORE DB command uses the device configuration file to read the database backup volumes. It is used by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

At recovery time, you may need to modify this stanza. You must update the device configuration information if the hardware configuration at the recovery site has changed. Examples of changes requiring updates to the configuration information are:

- Different device names
- Use of a manual library instead of an automated library
- For automated libraries, the requirement to manually place the database backup volumes in the automated library and update the configuration information to identify the element within the library. This allows the server to locate the required database backup volumes.

For details, see “Updating the Device Configuration File” on page 518.

The following rules determine where the device configuration file is placed at restore time:

- If the server options file contains DEVCONFIG entries, the server uses the fully qualified file name associated with the first entry. If the specified file name does not begin with a directory specification (for example, ‘.’ or ‘/’), the server adds the prefix *devcprefix*.
- If the server options file does not contain DEVCONFIG entries, the server uses the default name *devcprefix* followed by *drmdevc.txt*. For example, if *devcprefix* is /usr/tivoli/tsm/server/bin, the file name used by PREPARE is /usr/tivoli/tsm/server/bindrmdevc.txt.

Note: The *devcprefix* is set based on the following:

- If the environmental variable DSMSERV_DIR has been defined, it is used as the *devcprefix*.

- If the environmental variable DSMSERV_DIR has not been defined, the directory where the server is started from is used as the *devcprefix*.

If a fully qualified file name was not specified for the DEVCONFIG option in the server options file, the server adds it to the stanza DSMSERV.OPT.FILE.

```
begin DEVICE.CONFIGURATION.FILE

/* Tivoli Storage Manager Device Configuration */
DEFINE DEVCLASS COOL8MM DEVTYPE=8MM FORMAT=DRIVE MOUNTLIMIT=1 MOUNTWAIT=60
MOUNTRETENTION=60 PREFIX=TIVSM LIBRARY=ITSML
DEFINE DEVCLASS FILES DEVTYPE=FILE MAXCAPACITY=4096K MOUNTLIMIT=2 +
DIRECTORY=/usr/tivoli/tsm/server/bin
DEFINE DEVCLASS FILESSM DEVTYPE=FILE MAXCAPACITY=100K MOUNTLIMIT=2 +
DIRECTORY=/usr/tivoli/tsm/server/bin
DEFINE DEVCLASS LIB8MM DEVTYPE=8MM FORMAT=DRIVE MOUNTLIMIT=1 MOUNTWAIT=60+
MOUNTRETENTION=60 PREFIX=TIVSM LIBRARY=RLLIB
end DEVICE.CONFIGURATION.FILE
```

Figure 109. Device Configuration File Stanza

DSMSERV.OPT.FILE

Contains a copy of the server options file. This stanza is used by the RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE script.

Note: The following figure contains text strings that are too long to display in hardcopy or softcopy publications. The long text strings have a plus symbol (+) at the end of the string to indicate that they continue on the next line.

The disaster recovery plan file adds the DISABLESCHEDS option to the server options file and sets it to YES. This option disables administrative and client schedules while the server is being recovered. After the server is recovered, you can enable scheduling by deleting the option or setting it to NO and then restarting the server.

```
begin DSMSERV.OPT.FILE

* Server options file located in /usr/tivoli/tsm/server/bindsmserv.optx
TCPPort 1509
VOLUMEHISTORY /usr/tivoli/tsm/server/binvolhistory.txtx
DEVCONFIG     /usr/tivoli/tsm/server/bindevconfig.txtx
* The following option was added by PREPARE.
DISABLESCHEDS YES

end DSMSERV.OPT.FILE
```

Figure 110. Server Options File Stanza

License Information Stanza

LICENSE.INFORMATION
Contains a copy of the latest license audit results and the server license terms.

```
begin LICENSE.INFORMATION
                               Last License Audit: 12/30/2000 10:25:34
                          Registered Client Nodes: 1
                            Licensed Client Nodes: 51
                 Are network connections in use ?: Yes
               Are network connections licensed ?: Yes
Are Open Systems Environment clients registered ?: No
  Are Open Systems Environment clients licensed ?: No
                     Is space management in use ?: No
                   Is space management licensed ?: No
            Is disaster recovery manager in use ?: Yes
          Is disaster recovery manager licensed ?: Yes
    Are Server-to-Server Virtual Volumes in use ?: No
  Are Server-to-Server Virtual Volumes licensed ?: Yes
            Is Advanced Device Support required ?: No
            Is Advanced Device Support licensed ?: No
                        Server License Compliance: Valid

end LICENSE.INFORMATION
```

Figure 111. License Information Stanza

Machine Files Stanza

MACHINE.GENERAL.INFORMATION

Provides information for the server machine (for example, machine location). This stanza is included in the plan file if the machine information is saved in the database using the DEFINE MACHINE with ADSMSERVER=YES.

```
begin MACHINE.GENERAL.INFORMATION
Purpose: General information for machine DSMSRV1.
          This is the machine that contains DSM server DSM.
         Machine Name: DSMSRV1
     Machine Priority: 1
             Building: 21
                Floor: 2
                 Room: 2749
          Description: DSM Server for Branch 51
  Recovery Media Name: DSMSRVIMAGE

end MACHINE.GENERAL.INFORMATION
```

Figure 112. Machine General Information Stanza

MACHINE.RECOVERY.INSTRUCTIONS

Provides the recovery instructions for the server machine. This stanza is included in the plan file if the machine recovery instructions are saved in the database.

```
begin MACHINE.RECOVERY.INSTRUCTIONS
 Purpose: Recovery instructions for machine DSMSRV1.

Primary Contact:
   Jane Smith (wk 520-000-0000 hm 520-001-0001)
Secondary Contact:
   John Adams (wk 520-000-0001 hm 520-002-0002)

end MACHINE.RECOVERY.INSTRUCTIONS
```

Figure 113. Machine Recovery Instructions Stanza

MACHINE.RECOVERY.CHARACTERISTICS

Provides the hardware and software characteristics for the server machine.This stanza is included in the plan file if the machine characteristics are saved in the database.

```
begin MACHINE.CHARACTERISTICS
Purpose: Hardware and software characteristics of machine DSMSRV1.

  devices
  aio0         Defined                Asynchronous I/O
  bbl0         Available 00-0J        GXT150 Graphics Adapter
  bus0         Available 00-00        Microchannel Bus
  DSM1509bk02  Available              N/A
  DSM1509db01x Available              N/A
  DSM1509lg01x Available              N/A
  en0          Defined                Standard Ethernet Network Interface

end MACHINE.CHARACTERISTICS
```

Figure 114. Machine Recovery Characteristics Stanza

MACHINE.RECOVERY.MEDIA

Provides information about the media (for example, boot media) needed for rebuilding the machine that contains the server. This stanza is included in the plan file if recovery media information is saved in the database and it has been associated with the machine that contains the server.

```
begin MACHINE.RECOVERY.MEDIA.REQUIRED
 Purpose: Recovery media for machine DSMSRV1.
  Recovery Media Name: DSMSRVIMAGE
                 Type: Boot
         Volume Names: mkssy1
             Location: IRONMNT
          Description: mksysb image of server machine base OS
              Product: mksysb
  Product Information: this mksysb was generated by AIX 4.3

end MACHINE.RECOVERY.MEDIA.REQUIRED
```

Figure 115. Machine Recovery Media Stanza

Part 6. Appendixes

© Copyright IBM Corp. 1993, 2002

Appendix A. External Media Management Interface Description

This appendix contains Programming Interface information for the interface that Tivoli Storage Manager provides to external media management programs. The interface consists of request description strings that Tivoli Storage Manager sends and response strings that the external program sends.

To use the interface, you must first define an EXTERNAL-type TSM library that represents the media manager. You do not define drives, label volumes, or check in media. See “Configuring Libraries Controlled by Media Manager Programs” on page 88. Refer to your media manager’s documentation for that product’s setup information.

The details of the request types and the required processing are described in the sections that follow. The request types are:

- Initialization of the external program
- Begin Batch
- End Batch
- Volume Query
- Volume Eject
- Volume Release
- Volume Mount
- Volume Dismount

The responses can be right-padded with any number of white-space characters.

The *libraryname* passed in a request must be returned in the response. The *volume* specified in an eject request or a query request must be returned in the response. The *volume* specified in a mount request (except for ’SCRTCH’) must be returned in the response. When ’SCRTCH’ is specified in a mount request, the actual volume mounted must be returned.

CreateProcess Call

The server creates two anonymous uni-directional pipes and maps them to **stdin** and **stdout** during the **CreateProcess** call. According to Microsoft Developer Network documentation, if a standard handle has been redirected to refer to a file or a pipe, the handle can only be used by the ReadFile and WriteFile functions. This precludes normal C functions such as **gets** or **printf**. Since the server will never terminate the external program process, it is imperative that the external program recognize a read or write failure on the pipes and exit the process. In addition, the external program should exit the process if it reads an unrecognized command.

The external program may obtain values for the read and write handles using the following calls:

```
readPipe=GetStdHandle(STD_INPUT-HANDLE) and writePipe=GetStdHandle(STD_OUTPUT_HANDLE)
```

© Copyright IBM Corp. 1993, 2002

Processing during Server Initialization

Ensure that the external media management program cooperates with the server during the server's initialization. For each external library defined to the server, the following must occur during server initialization:

1. The server loads the external program (**CreateProcess**) in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string, in text form, into the standard input (**stdin**) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string, in text form, into its standard output (**stdout**) stream.
4. The server closes the pipes.
5. When the agent detects that the pipes are closed, it performs any necessary cleanup and calls the **stdlib** exit routine.

Processing for Mount Requests

To process the mount request:

1. The server loads the external program in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string (in text form) into the standard input (**stdin**) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string (in text form) into its standard output (**stdout**) stream.
4. The server sends the MOUNT request (**stdin**).
5. The agent sends the MOUNT response (**stdout**).
6. The agent waits.
7. The server sends the DISMOUNT request (**stdin**).
8. The agent sends the DISMOUNT response (**stdout**), performs any necessary cleanup, and calls the **stdlib** exit routine.

Processing for Release Requests

To process the release request:

1. The server loads the external program in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string (in text form) into the standard input (**stdin**) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string (in text form) into its standard output (**stdout**) stream.
4. The server sends the RELEASE request (**stdin**).
5. The agent sends the RELEASE response (**stdout**), performs any necessary cleanup, and calls the **stdlib** exit routine.

Processing for Batch Requests

Batch processing is done during MOVE MEDIA, MOVE DRMEDIA, and QUERY MEDIA command execution when performed on volumes in external libraries. The move commands will cause a QUERY to be issued for a volume. If the QUERY indicates that the volume is in the library, a subsequent EJECT for that volume is issued. As the move commands can match any number of volumes, a QUERY and an EJECT request is issued for each matching volume.

The QUERY MEDIA command will result in QUERY requests being sent to the agent. During certain types of processing, TSM may need to know if a volume is present in a library. The external agent should verify that the volume is physically present in the library.

1. The server loads the external program in a newly created process and creates pipes to the external program.
2. The server sends an initialization request description string (in text form) into the standard input (**stdin**) stream of the external program. The server waits for the response.
3. When the external process completes the request, the process must write an initialization response string (in text form) into its standard output (**stdout**) stream.
4. The server sends the BEGIN BATCH request (**stdin**).
5. The agent sends the BEGIN BATCH response (**stdout**).
6. The server sends 1 to n volume requests (n > 1). These can be any number of QUERY or EJECT requests. For each request, the agent will send the applicable QUERY response or EJECT response.
7. The server sends the END BATCH request (**stdin**).
8. The agent sends the END BATCH response (**stdout**), performs any necessary cleanup, and calls the **stdlib** exit routine.

Error Handling

If the server encounters an error during processing, it will close the **stdin** and **stdout** streams to the agent exit. The agent will detect this when it tries to read from **stdin** or write to **stdout**. If this occurs, the agent performs any necessary cleanup and calls the **stdlib** exit routine.

If the code for any response (except for EJECT and QUERY) is not equal to SUCCESS, TSM does not proceed with the subsequent steps. After the agent sends a non-SUCCESS return code for any response, the agent will perform any necessary cleanup and call the **stdlib** exit routine.

However, even if the code for EJECT or QUERY requests is not equal to SUCCESS, the agent will continue to send these requests.

If the server gets an error while trying to write to the agent, it will close the pipes, perform any necessary cleanup, and terminate the current request.

Begin Batch Request

The format of the Begin Batch Request is:

```
BEGIN BATCH
```

Format of the external program response:

```
BEGIN BATCH COMPLETE, RESULT=resultCode
```

where:

resultCode

One of the following:

- SUCCESS
- INTERNAL_ERROR

End Batch Request

The End Batch Request is sent by TSM to indicate that no more requests are to be sent by the external library manager for the current process. The external agent must send the End Batch Response and end by using the **stdlib** exit routine.

The format of the End Batch Request is:

```
END BATCH
```

Format of the external program response:

```
END BATCH COMPLETE, RESULT=resultCode
```

where:

resultCode

One of the following:

- SUCCESS
- INTERNAL_ERROR

Volume Query Request

The format of the Volume Query Request is:

```
QUERY libraryname volume
```

where:

libraryname

Specifies the name of the EXTERNAL library as defined to TSM.

volume

Specifies the volume name to be queried.

Format of the external program response:

```
QUERY libraryname volume COMPLETE, STATUS=statusValue, RESULT=resultCode
```

where:

libraryname

Specifies the name of the EXTERNAL library as defined to TSM.

volume

Specifies the volume name queried.

resultCode
One of the following:
- SUCCESS
- LIBRARY_ERROR
- VOLUME_UNKNOWN
- VOLUME_UNAVAILABLE
- CANCELLED
- TIMED_OUT
- INTERNAL_ERROR

If *resultCode* is not SUCCESS, the exit must return *statusValue* set to UNDEFINED. If *resultCode* is SUCCESS, STATUS must be one of the following values:
- IN_LIBRARY
- NOT_IN_LIBRARY

IN_LIBRARY means that the volume is currently in the library and available to be mounted.

NOT_IN_LIBRARY means that the volume is not currently in the library.

Initialization Requests

When the server is started, the server sends an initialization request to the external media management program for each EXTERNAL library. The external program must process this request to ensure that the external program is present, functional, and ready to process requests. If the initialization request is successful, TSM informs its operators that the external program reported its readiness for operations. Otherwise, TSM reports a failure to its operators.

TSM does not attempt any other type of operation with that library until an initialization request has succeeded. The server sends an initialization request first. If the initialization is successful, the request is sent. If the initialization is not successful, the request fails. The external media management program can detect whether the initialization request is being sent by itself or with another request by detecting end-of-file on the **stdin** stream. When end-of-file is detected, the external program must end by using the **stdlib** exit routine (not the **return** call).

When a valid response is sent by the external program, the external program must end by using the **exit** routine.

Format of the request:

```
INITIALIZE libraryname
```

where *libraryname* is the name of the EXTERNAL library as defined to TSM.

Format of the external program response:

```
INITIALIZE libraryname COMPLETE, RESULT=resultcode
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to TSM.

resultcode
One of the following:
- SUCCESS
- NOT_READY

- INTERNAL_ERROR

Volume Eject Request

The format of the Volume Eject Request is:

```
EJECT libraryname volume 'location info'
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to TSM.

volume
Specifies the volume to be ejected.

'location info'
Specifies the location information associated with the volume from the TSM inventory. It is delimited with single quotation marks. This information is passed without any modification from the TSM inventory. The customer is responsible for setting its contents with the appropriate UPDATE MEDIA or UPDATE VOLUME command before the move command is invoked. Set this field to some target location value that will assist in placing the volume after it is ejected from the library. It is suggested that the external agent post the value of this field to the operator.

Format of the external program response:

```
EJECT libraryname volume COMPLETE, RESULT=resultCode
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to TSM.

volume
Specifies the ejected volume.

resultCode
One of the following:

- SUCCESS
- LIBRARY_ERROR
- VOLUME_UNKNOWN
- VOLUME_UNAVAILABLE
- CANCELLED
- TIMED_OUT
- INTERNAL_ERROR

Volume Release Request

When the server returns a volume to scratch status, the server starts the external media management program, issues a request to initialize, then issues a request to release a volume.

The external program must send a response to the release request. No matter what response is received from the external program, TSM returns the volume to scratch. For this reason, TSM and the external program can have conflicting information on which volumes are scratch. If an error occurs, the external program should log the failure so that the external library inventory can be synchronized later with TSM. The synchronization can be a manual operation.

Format of the request:

```
RELEASE libraryname volname
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to TSM.

volname
Specifies the name of the volume to be returned to scratch (released).

Format of the external program response:

```
RELEASE libraryname volname COMPLETE, RESULT=resultcode
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to TSM.

volname
Specifies the name of the volume returned to scratch (released).

resultcode
One of the following:

- SUCCESS
- VOLUME_UNKNOWN
- VOLUME_UNAVAILABLE
- INTERNAL_ERROR

Volume Mount Request

When the server requires a volume mount, the server starts the external media management program, issues a request to initialize, then issues a request to mount a volume. The external program is responsible for verifying that this request is coming from TSM and not from an unauthorized system.

The volume mounted by the external media management program must be a tape with a standard IBM label that matches the external volume label. When the external program completes the mount request, the program must send a response. If the mount was successful, the external program must remain active. If the mount failed, the external program must end immediately by using the **stdlib** exit routine.

Format of the request:

```
MOUNT libraryname volname accessmode devicetypes timelimit userid
volumenumber 'location'
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to TSM.

volname
Specifies the actual volume name if the request is for an existing volume. If a scratch mount is requested, the *volname* is set to SCRTCH.

accessmode
Specifies the access mode required for the volume. Possible values are READONLY and READWRITE.

devicetypes

Specifies a list of device types that can be used to satisfy the request for the volume and the FORMAT specified in the device class. The most preferred device type is first in the list. Items are separated by commas, with no intervening spaces. Possible values are:

- 3480
- 3480XF
- 3490E
- 3570
- 3590
- 3590E
- 4MM_DDS1
- 4MM_DDS1C
- 4MM_DDS2
- 4MM_DDS2C
- 4MM_DDS3
- 4MM_DDS3C
- 4MM_DDS4
- 4MM_DDS4C
- 4MM_HP_DDS4
- 4MM_HP_DDS4C
- 8MM_8200
- 8MM_8205
- 8MM_8500
- 8MM_8500C
- 8MM_8900
- 8MM_AIT
- 8MM_AITC
- 8MM_ELIANT
- 8MM_M2
- DLT_2000
- DLT_4000
- DLT_7000
- SDLT
- DLT_8000
- DTF2
- DTF
- GENERICTAPE
- IBM_QIC4GBC
- LTO_ULTRIUM
- OPT_RW_650MB
- OPT_RW_1300MB
- OPT_RW_2600MB
- OPT_RW_5200MB
- OPT_WORM_650MB
- OPT_WORM_1300MB
- OPT_WORM12_5600MB
- OPT_WORM12_12000MB
- OPT_WORM14_14800MB
- QIC_12GBC
- QIC_20GBC
- QIC_25GBC
- QIC_30GBC
- QIC_50GBC
- QIC_IBM1000
- QIC_525

- QIC_5010C
- REMOVABLEFILE
- STK_9490
- STK_9840
- STK_9940
- STK_SD3

timelimit
Specifies the maximum number of minutes that the server waits for the volume to be mounted. If the mount request is not completed within this time, the external manager responds with the result code TIMED_OUT.

userid
Specifies the user ID of the process that needs access to the drive.

volumenumber
For non-optical media, the *volumenumber* is 1. For optical media, the *volumenumber* is 1 for side A, 2 for side B.

'location'
Specifies the value of the location field from the TSM inventory (for example, 'Room 617 Floor 2'). One blank character is inserted between the volume number and the left single quotation mark in the location information. If no location information is associated with a volume, nothing is passed to the exit. If no volume information exists, the single quotation marks are not passed. Also, if volume information is passed, then probably the volume has been ejected from the library and needs to be returned to the library before the mount operation can proceed. The location information should be posted by the agent so that the operator can obtain the volume and return it to the library.

Format of the external program response:

```
MOUNT libraryname volname COMPLETE ON specialfile, RESULT=resultcode
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to TSM.

volname
Specifies the name of the volume mounted for the request.

specialfile
The fully qualified path name of the device special file for the drive in which the volume was mounted. If the mount request fails, the value should be set to /dev/null.

The external program must ensure that the special file is closed before the response is returned to the server.

resultcode
One of the following:

- SUCCESS
- DRIVE_ERROR
- LIBRARY_ERROR
- VOLUME_UNKNOWN
- VOLUME_UNAVAILABLE
- CANCELLED
- TIMED_OUT
- INTERNAL_ERROR

Volume Dismount Request

When a successful mount operation completes, the external process must wait for a request to dismount the volume. When the dismount operation completes, the external program must send a response to the server.

After the dismount response is sent, the external process ends immediately by using the **stdlib** exit routine.

Format of the request:

```
DISMOUNT libraryname volname
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to TSM.

volname
Specifies the name of the volume to be dismounted.

Format of the external program response:

```
DISMOUNT libraryname volname COMPLETE, RESULT=resultcode
```

where:

libraryname
Specifies the name of the EXTERNAL library as defined to TSM.

volname
Specifies the name of the volume dismounted.

resultcode
One of the following:

- SUCCESS
- DRIVE_ERROR
- LIBRARY_ERROR
- INTERNAL_ERROR

Appendix B. User Exit and File Exit Receivers

This appendix contains samples of the user exit receiver for event logging. The data structure of the user exit receivers also applies to the file exit receivers. To use one of these exits with TSM, you must specify the corresponding server option (FILEEXIT, FILETEXTEXIT, or USEREXIT) in the server options file. You can also use TSM commands to control event logging. See "Logging Tivoli Storage Manager Events to Receivers" on page 413 and *Administrator's Reference* for details. The samples for the C, H, and make files are shipped with the server code in the */usr/lpp/adsmserv/bin* directory.

Notes:

1. Use caution in modifying these exits. A user exit abend will bring down the server.
2. The file specified in the file exit option will continue to grow unless you prune it.

© Copyright IBM Corp. 1993, 2002

Sample User Exit Declarations

```
/*****************************************************************
 * Name:            userExitSample.h
 * Description:     Declarations for a user-exit
 * Environment:     AIX 4.1.4+ on RS/6000
 *****************************************************************/

#ifndef _H_USEREXITSAMPLE
#define _H_USEREXITSAMPLE

#include <stdio.h>
#include <sys/types.h>

/*****  Do not modify below this line.  *****/

#define BASE_YEAR       1900

typedef short int16;
typedef int int32;

/* uchar is usually defined in <sys/types.h> */
/* DateTime Structure Definitions - TSM representation of a timestamp*/

typedef struct
{
  uchar   year; /* Years since BASE_YEAR (0-255) */
  uchar mon;     /* Month (1 - 12)  */
  uchar    day;     /* Day (1 - 31)  */
  uchar hour; /* Hour (0 - 23)  */
  uchar min;   /* Minutes (0 - 59)  */
  uchar sec;   /* Seconds (0 - 59)  */
} DateTime;

/******************************************
 * Some field size definitions (in bytes) *
 ******************************************/

#define MAX_SERVERNAME_LENGTH   64
#define MAX_NODE_LENGTH   64
#define MAX_COMMNAME_LENGTH   16
#define MAX_OWNER_LENGTH   64
#define MAX_HL_ADDRESS   64
#define MAX_LL_ADDRESS   32
#define MAX_SCHED_LENGTH   30
#define MAX_DOMAIN_LENGTH   30
#define MAX_MSGTEXT_LENGTH 1600
```

Figure 116. Sample User Exit Declarations (Part 1 of 3)

```
/**********************************************
 * Event Types (in elEventRecvData.eventType) *
 **********************************************/

#define TSM_SERVER_EVENT       0x03  /* Server Events */
#define TSM_CLIENT_EVENT       0x05  /* Client Events */

/***************************************************
 * Application Types (in elEventRecvData.applType) *
 ***************************************************/

#define TSM_APPL_BACKARCH    1  /* Backup or Archive client      */
#define TSM_APPL_HSM         2  /* Space manage client           */
#define TSM_APPL_API         3  /* API client                    */
#define TSM_APPL_SERVER      4  /* Server (ie. server to server )*/

/*****************************************************
 * Event Severity Codes (in elEventRecvData.sevCode) *
 *****************************************************/

#define TSM_SEV_INFO         0x02     /* Informational message.   */
#define TSM_SEV_WARNING      0x03     /* Warning message.
     */
#define TSM_SEV_ERROR        0x04     /* Error message.           */
#define TSM_SEV_SEVERE       0x05     /* Severe error message.    */
#define TSM_SEV_DIAGNOSTIC   0x06     /* Diagnostic message.      */
#define TSM_SEV_TEXT         0x07     /* Text message.            */

/************************************************************
 * Data Structure of Event that is passed to the User-Exit. *
 * This data structure is the same for a file generated via *
 *    FILEEXIT option on the server.                        *
 ************************************************************/

typedef struct evRdata
{
  int32    eventNum;              /* the event number.               */
  int16    sevCode;               /* event severity.                 */
  int16    applType;              /* application type (hsm, api, etc)*/
  int32    sessId;                /* session number                    */
  int32    version;               /* Version number of this structure (1)*/
  int32    eventType;             /* event type                            *
                                  * (TSM_CLIENT_EVENT, TSM_SERVER_EVENT)*/
```

Figure 116. Sample User Exit Declarations (Part 2 of 3)

```
  DateTime timeStamp;           /* timestamp for event data.           */
  uchar    serverName[MAX_SERVERNAME_LENGTH+1]; /* server name         */
  uchar    nodeName[MAX_NODE_LENGTH+1]; /* Node name for session       */
  uchar    commMethod[MAX_COMMNAME_LENGTH+1]; /* communication method  */
  uchar    ownerName[MAX_OWNER_LENGTH+1];     /* owner                 */
  uchar    hlAddress[MAX_HL_ADDRESS+1];       /* high-level address    */
  uchar    llAddress[MAX_LL_ADDRESS+1];       /*  low-level address    */
  uchar    schedName[MAX_SCHED_LENGTH+1]; /* schedule name if applicable*/
  uchar    domainName[MAX_DOMAIN_LENGTH+1]; /* domain name for node    */
  uchar    event[MAX_MSGTEXT_LENGTH];       /* event text              */
} elEventRecvData;

/************************************
 * Size of the Event data structure *
 ************************************/

#define ELEVENTRECVDATA_SIZE            sizeof(elEventRecvData)

/*************************************
 * User Exit EventNumber for Exiting *
 *************************************/

#define USEREXIT_END_EVENTNUM     1822  /* Only user-exit receiver to exit*/
#define END_ALL_RECEIVER_EVENTNUM 1823  /* All receivers told to exit   */

/**************************************
 *** Do not modify above this line. ***
 **************************************/

/********************** Additional Declarations **************************/

#endif
```

Figure 116. Sample User Exit Declarations (Part 3 of 3)

Sample User Exit Program

```
/***********************************************************************
 * Name:            userExitSample.c
 * Description:     Example user-exit program invoked by the TSM V3 Server
 * Environment:     AIX 4.1.4+ on RS/6000
 ***********************************************************************/

#include <stdio.h>
#include "userExitSample.h"

/**************************************
 *** Do not modify below this line. ***
 **************************************/

extern void adsmV3UserExit( void *anEvent );

/************
 *** Main ***
 ************/

int main(int argc, char *argv[])
{
/* Do nothing, main() is never invoked, but stub is needed */

exit(0);  /* For picky compilers */

} /* End of main() */

/******************************************************************
 * Procedure:  adsmV3UserExit
 * If the user-exit is specified on the server, a valid and
 * appropriate event causes an elEventRecvData structure (see
 * userExitSample.h) to be passed to adsmV3UserExit that returns a void.
 * INPUT :   A (void *) to the elEventRecvData structure
 * RETURNS:  Nothing
 ******************************************************************/

void adsmV3UserExit( void *anEvent )
{
/* Typecast the event data passed */
elEventRecvData *eventData = (elEventRecvData *)anEvent;
```

Figure 117. Sample User Exit Program (Part 1 of 2)

```
 /*************************************
 *** Do not modify above this line. ***
 *************************************/

if( ( eventData->eventNum == USEREXIT_END_EVENTNUM     ) ||
    ( eventData->eventNum == END_ALL_RECEIVER_EVENTNUM ) )
  {
   /* Server says to end this user-exit.  Perform any cleanup, *
    * but do NOT exit() !!!                                    */
   return;
  }

/* Field Access:  eventData->.... */
/* Your code here ... */

return; /* For picky compilers */
} /* End of adsmV3UserExit() */
```

Figure 117. Sample User Exit Program (Part 2 of 2)

Readable Text File Exit (FILETEXTEXIT) Format

If you specify the readable text file exit (FILETEXTEXIT), each logged event is written to a fixed-size, readable line. The following table presents the format of the output. Fields are separated by blank spaces.

Table 36. Readable Text File Exit (FILETEXTEXIT) Format

Column	Description
0001-0006	Event number (with leading zeros)
0008-0010	Severity code number
0012-0013	Application type number
0015-0023	Session ID number
0025-0027	Event structure version number
0029-0031	Event type number
0033-0046	Date/Time (YYYYMMDDDHHmmSS)
0048-0111	Server name (right padded with spaces)
0113-0176	Node name
0178-0193	Communications method name
0195-0258	Owner name
0260-0323	High-level internet address (n.n.n.n)
0325-0356	Port number from high-level internet address
0358-0387	Schedule name
0389-0418	Domain name
0420-2019	Event text
2020-2499	Unused spaces
2500	New line character

Appendix C. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corp. 1993, 2002

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Information Enabling Requests
Dept. M13
5600 Cottle Road
San Jose CA 95193-0001
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface

This publication is intended to help the customer plan for and manage the Tivoli Storage Manager server.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Tivoli Storage Manager. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Programming interface information

End of Programming interface information

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States or other countries or both:

Advanced Peer-to-Peer Networking
AIX
Application System/400
APPN
AS/400
DB2
DFDSM
DFS
DFSMS/MVS
DFSMShsm
DFSMSrmm
DPI
Enterprise Storage Server
ESCON
Extended Services
FlashCopy
IBM
IBMLink
Magstar
MVS
MVS/ESA
MVS/SP
NetView
OpenEdition
Operating System/2
Operating System/400
OS/2
OS/390
OS/400
POWERparallel
RACF
Redbooks
RISC System/6000
RS/6000
SAA
SANergy
SP
System/370
System/390
SystemView
Tivoli
Tivoli Enterprise Console
Tivoli Management Environment
TME
VTAM
z/OS

Lotus, Lotus 1–2–3, Lotus Approach, Lotus Domino and Lotus Notes are trademarks of Lotus Development Corporation in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark of the Open Group in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Intel is a registered trademark of the Intel Corporation in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

The terms in this glossary are defined as they pertain to the Tivoli Storage Manager library. If you do not find the term you need, refer to the IBM Software Glossary on the Web at this URL: www.ibm.com/ibm/terminology/. You can also refer to *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

This glossary may include terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York 10036.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC2/SC1).

A

absolute mode. A backup copy group mode indicating that specifies that a file is considered for incremental backup even if the file has not changed since the last backup. See also *mode*. Contrast with *modified mode*.

access mode. An attribute of a storage pool or a storage volume that specifies whether the server can write to or read from the storage pool or storage volume. The access mode can be read/write, read-only, or unavailable. Volumes in primary storage pools can also have an access mode of destroyed. Volumes in copy storage pools can also have an access mode of offsite.

activate. To validate the contents of a policy set and make it the active policy set.

active policy set. The activated policy set that contains the policy rules currently in use by all client nodes assigned to the policy domain. See also *policy domain* and *policy set*.

active version. The most recent backup copy of a file stored by Tivoli Storage Manager. The active version of a file cannot be deleted until a backup process detects that the user has either replaced the file with a newer version or has deleted the file from the workstation. Contrast with *inactive version*.

activity log. A log that records normal activity messages generated by the server. These messages include information about server and client operations, such as the start time of sessions or device I/O errors. Each message includes a message ID, date and time stamp, and a text description. The number of days to retain messages in the activity log can be specified.

administrative client. A program that runs on a file server, workstation, or mainframe that administrators use to control and monitor the Tivoli Storage Manager server. Contrast with *backup-archive client.*

administrative command schedule. A database record that describes the planned processing of an administrative command during a specific time period. See also *client schedule*.

administrative privilege class. See *privilege class*.

administrative session. A period of time in which an administrator user ID communicates with a server to perform administrative tasks. Contrast with *client node session*.

administrator. A user who has been registered to the server. Administrators can be authorized to one or more of the following administrative privilege classes: system, policy, storage, operator, or analyst. Administrators can use the administrative commands and queries allowed by their privileges.

aggregate file. A file, stored in one or more storage pools, consisting of a group of logical files packaged together. See *logical file* and *physical file*.

analyst privilege class. A privilege class that allows an administrator to reset statistics. See also *privilege class*.

application client. One of the Tivoli Data Protection programs installed on a system to protect an application. The Tivoli Storage Manager server provides backup services to an application client.

archive. To copy one or more files to a storage pool for long-term storage. Archived files can include descriptive information and can be retrieved by archive date, by file name, or by description. Contrast with *retrieve*.

archive copy. A file that has been archived to server storage.

archive copy group. A policy object containing attributes that control the generation, destination, and

© Copyright IBM Corp. 1993, 2002

expiration of archive copies. An archive copy group belongs to a management class.

archive retention grace period. The number of days that Tivoli Storage Manager retains an archive copy when the server is unable to rebind the file to an appropriate management class.

assigned capacity. The portion of available space that can be used to store database or recovery log information. See also *available space*.

association. (1) The defined relationship between a client node and a client schedule. An association identifies the name of a schedule, the name of the policy domain to which the schedule belongs, and the name of a client node that performs scheduled operations. (2) On a configuration manager, the defined relationship between a profile and an object such as a policy domain. Profile associations define the configuration information that will be distributed to a managed server when it subscribes to the profile.

audit. To check for logical inconsistencies between information that the server has and the actual condition of the system. Tivoli Storage Manager can audit volumes, the database, libraries, and licenses. For example, when Tivoli Storage Manager audits a volume, the server checks for inconsistencies between information about backed-up or archived files stored in the database and the actual data associated with each backup version or archive copy in server storage.

authentication. The process of checking a user's password before allowing that user access to the server. Authentication can be turned on or off by an administrator with system privilege.

authority. The right granted to a user to perform tasks with Tivoli Storage Manager servers and clients. See also *privilege class*.

autochanger. A small, multislot tape device that automatically puts tape cartridges into tape drives. See also *library*.

available space. The amount of space, in megabytes, that is available to the database and recovery log. This space can be used to extend the capacity of the database or recovery log, or to provide sufficient free space before a volume is deleted from the database or recovery log.

B

back up. To copy information to another location to ensure against loss of data. In Tivoli Storage Manager, you can back up user files, the Tivoli Storage Manager database, and storage pools. Contrast with *restore*. See also *database backup series* and *incremental backup*.

backup-archive client. A program that runs on a workstation or file server and provides a means for users to back up, archive, restore, and retrieve files. Contrast with *administrative client*.

backup copy group. A policy object containing attributes that control the generation, destination, and expiration of backup versions of files. A backup copy group belongs to a management class.

backup retention grace period. The number of days that Tivoli Storage Manager retains a backup version after the server is unable to rebind the file to an appropriate management class.

backup set. A portable, consolidated group of active backup versions of files, generated for a backup-archive client.

backup version. A file that a user backed up to server storage. More than one backup version can exist in server storage, but only one backup version is the active version. See also *active version* and *inactive version*.

binding. The process of associating a file with a management class name. See *rebinding*.

buffer pool. Temporary space used by the server to hold database or recovery log pages. See *database buffer pool* and *recovery log buffer pool*.

C

cache. The process of leaving a duplicate copy on random access media when the server migrates a file to another storage pool in the hierarchy.

central scheduler. A function that allows an administrator to schedule client operations and administrative commands. The operations can be scheduled to occur periodically or on a specific date. See *client schedule* and *administrative command schedule*.

client. A program running on a PC, workstation, file server, LAN server, or mainframe that requests services of another program, called the server. The following types of clients can obtain services from a Tivoli Storage Manager server: administrative client, application client, API client, backup-archive client, and HSM client (also known as Tivoli Space Manager).

client domain. The set of drives, file systems, or volumes that the user selects to back up or archive using the backup-archive client.

client migration. The process of copying a file from a client node to server storage and replacing the file with a stub file on the client node. The space management attributes in the management class control this migration. See also *space management*.

client node. A file server or workstation on which the backup-archive client program has been installed, and which has been registered to the server.

client node session. A period of time in which a client node communicates with a server to perform backup, restore, archive, retrieve, migrate, or recall requests. Contrast with *administrative session*.

client options file. A file that a client can change, containing a set of processing options that identify the server, communication method, and options for backup, archive, hierarchical storage management, and scheduling. Also called the *dsm.opt* file.

client-polling scheduling mode. A client/server communication technique where the client queries the server for work. Contrast with *server-prompted scheduling mode*.

client schedule. A database record that describes the planned processing of a client operation during a specific time period. The client operation can be a backup, archive, restore, or retrieve operation, a client operating system command, or a macro. See also *administrative command schedule*.

client system options file. A file, used on UNIX clients, containing a set of processing options that identify the Tivoli Storage Manager servers to be contacted for services. This file also specifies communication methods and options for backup, archive, hierarchical storage management, and scheduling. Also called the *dsm.sys* file. See also *client user options file*.

client user options file. A user-created file, used on UNIX clients, containing a set of processing options that identify the server, communication method, backup and archive options, space management options, and scheduling options. Also called the *dsm.opt* file. See also *client system options file*.

closed registration. A registration process in which only an administrator can register workstations as client nodes with the server. Contrast with *open registration*.

collocation. The process of keeping all data belonging to a single client node or a single client file space on a minimal number of sequential-access volumes within a storage pool. Collocation can reduce the number of volumes that must be accessed when a large amount of data must be restored.

compression. The process of saving storage space by eliminating empty fields or unnecessary data in a file. In Tivoli Storage Manager, compression can occur at a workstation before files are backed up or archived to server storage. On some types of tape drives, hardware compression can be used.

configuration manager. One Tivoli Storage Manager server that distributes configuration information to other Tivoli Storage Manager servers (called managed servers) via profiles. Configuration information can include policy and schedules. See *managed server* and *profile*.

copy group. A policy object whose attributes control how backup versions or archive copies are generated, where backup versions or archive copies are initially located, and when backup versions or archive copies expire. A copy group belongs to a management class. See also *archive copy group*, *backup copy group*, *backup version*, and *management class*.

copy storage pool. A named set of volumes that contains copies of files that reside in primary storage pools. Copy storage pools are used only to back up the data stored in primary storage pools. A copy storage pool cannot be a destination for a backup copy group, an archive copy group, or a management class (for space-managed files). See *primary storage pool* and *destination*.

D

damaged file. A physical file for which Tivoli Storage Manager has detected read errors.

database. A collection of information about all objects managed by the server, including policy management objects, users and administrators, and client nodes.

database backup series. One full backup of the database, plus up to 32 incremental backups made since that full backup. Each full backup that is run starts a new database backup series. A backup series is identified with a number.

database backup trigger. A set of criteria that defines when and how database backups are run automatically. The criteria determine how often the backup is run, whether the backup is a full or incremental backup, and where the backup is stored.

database buffer pool. Storage that is used as a cache to allow database pages to remain in memory for long periods of time, so that the server can make continuous updates to pages without requiring input or output (I/O) operations from external storage.

database snapshot. A complete backup of the entire Tivoli Storage Manager to media that can be taken off-site. When a database snapshot is created, the current database backup series is not interrupted. A database snapshot cannot have incremental database backups associated with it. See also *database backup series*. Contrast with *full backup*.

data mover. A device, defined to Tivoli Storage Manager, that moves data on behalf of the server. A NAS file server can be a data mover.

default management class. A management class assigned to a policy set that the server uses to manage backed-up or archived files when a user does not specify a management class for a file.

desktop client. The group of backup-archive clients supported by Tivoli Storage Manager that includes clients on Windows, Apple, and Novell NetWare operating systems.

destination. A copy group or management class attribute that specifies the primary storage pool to which a client file will be backed up, archived, or migrated.

device class. A named set of characteristics applied to a group of storage devices. Each device class has a unique name and represents a device type of disk, file, optical disk, or tape.

device configuration file. A file that contains information about defined device classes, and, on some Tivoli Storage Manager servers, defined libraries and drives. The file can be created by using a Tivoli Storage Manager administrative command or by using an option in the server options file. The information is a copy of the device configuration information in the Tivoli Storage Manager database.

disaster recovery plan. A file created by Tivoli Disaster Recovery Manager (DRM) that contains information about how to recover computer systems if a disaster occurs and scripts that can be run to perform some recovery tasks. The file includes information about the software and hardware used by the Tivoli Storage Manager server and the location of recovery media.

domain. See *policy domain* or *client domain*.

DRM. A short name for Tivoli Disaster Recovery Manager.

dsm.opt file. See *client options file* and *client user options file*.

dsmserv.opt. See *server options file*.

dsm.sys file. See *client system options file*.

dynamic. A value for serialization that specifies that Tivoli Storage Manager accepts the first attempt to back up or archive a file regardless of whether the file is modified during the backup or archive process. See also *serialization*. Contrast with *shared dynamic*, *shared static*, and *static*.

E

enterprise configuration. A method of setting up Tivoli Storage Manager servers so that the administrator can distribute the configuration of one of the servers to the other servers, using server-to-server communication. See *configuration manager*, *managed server*, *profile*, and *subscription*.

enterprise logging. The sending of events from Tivoli Storage Manager servers to a designated event server. The event server routes the events to designated receivers, such as to a user exit. See also *event*.

estimated capacity. The available space, in megabytes, of a storage pool.

event. (1) An administrative command or a client operation that is scheduled to be run using Tivoli Storage Manager scheduling. (2) A message that a Tivoli Storage Manager server or client issues. Messages can be logged using Tivoli Storage Manager event logging.

event record. A database record that describes actual status and results for events.

event server. A server to which other servers can send events for logging. The event server routes the events to any receivers that are enabled for the sending server's events.

exclude. To identify files that you do not want to include in a specific client operation, such as backup or archive. You identify the files in an include-exclude list.

exclude-include list. See *include-exclude list*.

expiration. The process by which files are identified for deletion because their expiration date or retention period has passed. Backed-up or archived files are marked expired by Tivoli Storage Manager based on the criteria defined in the backup or archive copy group.

expiration date. On some Tivoli Storage Manager servers, a device class attribute used to notify tape management systems of the date when Tivoli Storage Manager no longer needs a tape volume. The date is placed in the tape label so that the tape management system does not overwrite the information on the tape volume before the expiration date.

export. To copy administrator definitions, client node definitions, policy definitions, server control information, or file data to external media. Used to move or copy information between servers.

extend. To increase the portion of available space that can be used to store database or recovery log information. Contrast with *reduce*.

F

file space. A logical space in Tivoli Storage Manager server storage that contains a group of files. For clients on Windows systems, a file space is a logical partition that is identified by a volume label. For clients on UNIX systems, a file space is a logical space that contains a group of files backed up or archived from the same file system, or part of a file system that stems from a virtual mount point. Clients can restore, retrieve, or delete their file spaces from Tivoli Storage Manager server storage. Tivoli Storage Manager does not necessarily store all the files from a single file space together, but can identify all the files in server storage that came from a single file space.

file space ID (FSID). A unique numeric identifier that the server assigns to a file space when it is stored in server storage.

frequency. A copy group attribute that specifies the minimum interval, in days, between incremental backups.

FSID. See *file space ID*.

full backup. The process of backing up the entire server database. A full backup begins a new database backup series. See also *database backup series* and *incremental backup*. Contrast with *database snapshot*.

fuzzy copy. A backup version or archive copy of a file that might not accurately reflect what the original content of the file because Tivoli Storage Manager backed up or archived the file while the file was being modified.

H

hierarchical storage management (HSM) client. The Tivoli Space Manager program that runs on workstations to allow users to maintain free space on their workstations by migrating and recalling files to and from Tivoli Storage Manager storage. Synonymous with *space manager client*.

high migration threshold. A percentage of the storage pool capacity that identifies when the server can start migrating files to the next available storage pool in the hierarchy. Contrast with *low migration threshold*. See *server migration*.

HSM client. Hierarchical storage management client. Also known as the space manager client.

I

image backup. A backup of a full file system or raw logical volume as a single object.

import. The process of copying exported administrator definitions, client node definitions, policy definitions, server control information or file data from external media to a target server. A subset of information can be imported to a target server from the external media. Used to move or copy information between servers. See *export*.

inactive version. A backup version of a file that is either not the most recent backup version or that is a backup version of a file that no longer exists on the client system. Inactive backup versions are eligible for expiration processing according to the management class assigned to the file. Contrast with *active version*.

include-exclude file. On UNIX and Windows clients, a file containing statements that Tivoli Storage Manager uses to determine whether to include certain files in specific client operations, and to determine the associated management classes to use for backup, archive, and space management. See *include-exclude list*.

include-exclude list. A group of include and exclude option statements that Tivoli Storage Manager uses. The exclude options identify files that are not to be included in specific client operations such as backup or space management. The include options identify files that are exempt from the exclusion rules. The include options can also assign a management class to a file or group of files for backup, archive, or space management services. The include-exclude list for a client may include option statements from the include-exclude file (for UNIX clients) or the client options file (for other clients), and from a client option set on the server.

incremental backup. (1) The process of backing up files or directories that are new or have changed since the last incremental backup. See also *selective backup*. (2) The process of copying only the pages in the database that are new or changed since the last full or incremental backup of the database. Contrast with *full backup*. See also *database backup series*.

L

LAN-free data transfer. The movement of client data directly between a client and a storage device over a SAN, rather than over the LAN.

library. (1) A repository for demountable recorded media, such as magnetic tapes. (2) For Tivoli Storage Manager, a collection of one or more drives, and possibly robotic devices (depending on the library type), which can be used to access storage volumes. (3) In the AS/400® system, a system object that serves as a directory to other objects. A library groups related objects, and allows the user to find objects by name.

library client. A Tivoli Storage Manager server that uses server-to-server communication to access a library that is managed by another Tivoli Storage Manager server. See also *library manager*.

library manager. A Tivoli Storage Manager server that controls device operations when multiple Tivoli Storage Manager servers share a storage device. The device operations include mount, dismount, volume ownership, and library inventory. See also *library client*.

logical file. A file stored in one or more server storage pools, either by itself or as part of an aggregate file. See also *aggregate file* and *physical file*.

logical occupancy. The amount of space used by logical files in a storage pool. This space does not include the unused space created when logical files are deleted from aggregate files, so it might be less than the physical occupancy. See also *physical occupancy*, *physical file*, and *logical file*.

logical volume. (1) A portion of a physical volume that contains a filesystem. (2) For the Tivoli Storage Manager server, the combined space on all volumes for either the database or the recovery log. The database is one logical volume and the recovery log is one logical volume.

low migration threshold. A percentage of the storage pool capacity that specifies when the server can stop the migration of files to the next storage pool. Contrast with *high migration threshold*. See *server migration*.

M

macro file. A file that contains one or more Tivoli Storage Manager administrative commands, which can be run only from an administrative client by using the MACRO command. Contrast with *Tivoli Storage Manager command script*.

managed object. A definition in the database of a managed server that was distributed to the managed server by a configuration manager. When a managed server subscribes to a profile, all objects associated with that profile become managed objects in the database of the managed server. In general, a managed object cannot be modified locally on the managed server. Objects can include policy, schedules, client options sets, server scripts, administrator registrations, and server and server group definitions.

managed server. A Tivoli Storage Manager server that receives configuration information from a configuration manager via subscription to one or more profiles. Configuration information can include definitions of objects such as policy and schedules. See *configuration manager*, *subscription*, and *profile*.

managed system. A client or server that requests services from the Tivoli Storage Manager server.

management class. A policy object that users can bind to each file to specify how the server manages the file. The management class can contain a backup copy group, an archive copy group, and space management attributes. The copy groups determine how the server manages backup versions or archive copies of the file. The space management attributes determine whether the file is eligible to be migrated by the space manager client to server storage and under what conditions the file is migrated. See also *copy group*, *space manager client*, *binding*, and *rebinding*.

maximum extension. Specifies the maximum amount of storage space, in megabytes, that you can extend the database or recovery log.

maximum reduction. Specifies the maximum amount of storage space, in megabytes, that you can reduce the database or recovery log.

maximum utilization. The highest percentage of assigned capacity used by the database or recovery log.

migrate. To move data from one storage location to another. See also *client migration* and *server migration*.

mirroring. The process of writing the same data to multiple disks at the same time. Mirroring data protects against data loss within the database or recovery log.

mode. A copy group attribute that specifies whether to back up a file that has not been modified since the last time the file was backed up. See *modified* and *absolute*.

modified mode. A backup copy group mode that specifies that a file is considered for incremental backup only if it has changed since the last backup. A file is considered changed if the date, size, owner, or permissions have changed. See also *mode*. Contrast with *absolute mode*.

mount. To place a data medium (such as a tape cartridge) on a drive in a position to operate.

mount limit. A device class attribute that specifies the maximum number of volumes that can be simultaneously accessed from the same device class. The mount limit determines the maximum number of mount points. See *mount point*.

mount point. A logical drive through which the server accesses volumes in a sequential access device class. For a removable media device such as tape, a mount point is a logical drive associated with a physical drive. For a device class with the device type of FILE, a mount point is a logical drive associated with an I/O stream. The number of mount points for a device class is determined by the mount limit for that class. See *mount limit*.

mount retention period. A device class attribute that specifies the maximum number of minutes that the server retains a mounted sequential access media volume that is not being used before it dismounts the sequential access media volume.

mount wait period. A device class attribute that specifies the maximum number of minutes that the server waits for a sequential access volume mount request to be satisfied before canceling the request.

N

NAS. Network-attached storage.

NAS node. A Tivoli Storage Manager node that is a NAS file server. Data for the NAS node is transferred by the NAS file server itself at the direction of a Tivoli Storage Manager server that uses NDMP. The data is not transferred by the Tivoli Storage Manager client. Also called NAS file server node.

native format. A format of data that is written to a storage pool directly by the Tivoli Storage Manager server. Contrast with *non-native data format*.

NDMP. Network Data Management Protocol.

network-attached storage (NAS) file server. A dedicated storage device with an operating system that is optimized for file-serving functions. In Tivoli Storage Manager, a NAS file server can have the characteristics of both a node and a data mover. See also *data mover* and *NAS node*.

Network Data Management Protocol (NDMP). An industry-standard protocol that allows a network storage-management application (such as Tivoli Storage Manager) to control the backup and recovery of an NDMP-compliant file server, without installing third-party software on that file server.

node. (1) A workstation or file server that is registered with a Tivoli Storage Manager server to receive its services. See also *client node* and *NAS node*. (2) In a Microsoft cluster configuration, one of the computer systems that make up the cluster.

node privilege class. A privilege class that allows an administrator to remotely access backup-archive clients for a specific client node or for all clients in a policy domain. See also *privilege class*.

non-native data format. A format of data written to a storage pool that is different from the format that the server uses for basic LAN-based operations. The data is written by a data mover instead of the server. Storage pools with data written in a non-native format may not support some server operations, such as audit of a volume. The NETAPPDUMP data format for NAS node backups is an example of a non-native data format.

O

open registration. A registration process in which any users can register their own workstations as client nodes with the server. Contrast with *closed registration*.

operator privilege class. A privilege class that allows an administrator to issue commands that disable or halt the server, enable the server, cancel server processes, and manage removable media. See also *privilege class*.

P

page. A unit of space allocation within Tivoli Storage Manager database volumes.

path. A Tivoli Storage Manager object that defines a one-to-one relationship between a source and a destination. Using the path, the source accesses the destination. Data may flow from the source to the destination, and back. An example of a source is a data mover (such as a NAS file server), and an example of a destination is a tape drive.

physical file. A file stored in one or more storage pools, consisting of either a single logical file, or a group of logical files packaged together (an aggregate file). See also *aggregate file* and *logical file*.

physical occupancy. The amount of space used by physical files in a storage pool. This space includes the unused space created when logical files are deleted from aggregates. See also *physical file*, *logical file*, and *logical occupancy*.

policy domain. A policy object that contains policy sets, management classes, and copy groups that are used by a group of client nodes. See *policy set*, *management class*, and *copy group*.

policy privilege class. A privilege class that allows an administrator to manage policy objects, register client nodes, and schedule client operations for client nodes. Authority can be restricted to certain policy domains. See also *privilege class*.

policy set. A policy object that contains a group of management classes that exist for a policy domain. Several policy sets can exist within a policy domain but only one policy set is active at one time. See *management class* and *active policy set*.

premigration. For a space manager client, the process of copying files that are eligible for migration to server storage, while leaving the original file intact on the local system.

primary storage pool. A named set of volumes that the server uses to store backup versions of files, archive copies of files, and files migrated from HSM client nodes. You can back up a primary storage pool to a copy storage pool. See *destination* and *copy storage pool*.

privilege class. A level of authority granted to an administrator. The privilege class determines which administrative tasks the administrator can perform. For example, an administrator with system privilege class can perform any administrative task. Also called administrative privilege class. See also system privilege class, policy privilege class, storage privilege class, operator privilege class, and analyst privilege class. See also *system privilege class*, *policy privilege class*, *storage* *privilege class*, *operator privilege class*, *analyst privilege class*, and *node privilege class*.

profile. A named group of configuration information that can be distributed from a configuration manager when a managed server subscribes. Configuration information can include registered administrators, policy, client schedules, client option sets, administrative schedules, Tivoli Storage Manager command scripts, server definitions, and server group definitions. See *configuration manager* and *managed server*.

R

randomization. The process of distributing schedule start times for different clients within a specified percentage of the schedule's startup window.

rebinding. The process of associating a backed-up file with a new management class name. For example, rebinding occurs when the management class associated with a file is deleted. See *binding*.

recall. To access files that have been migrated from workstations to server storage by using the space manager client. Contrast with *migrate*.

receiver. A server repository that contains a log of server messages and client messages as events. For example, a receiver can be a file exit, a user exit, or the Tivoli Storage Manager server console and activity log. See also *event*.

reclamation. A process of consolidating the remaining data from many sequential access volumes onto fewer new sequential access volumes.

reclamation threshold. The percentage of reclaimable space that a sequential access media volume must have before the server can reclaim the volume. Space becomes reclaimable when files are expired or are deleted. The percentage is set for a storage pool.

recovery log. A log of updates that are about to be written to the database. The log can be used to recover from system and media failures.

recovery log buffer pool. Storage that the server uses to hold new transaction records until they can be written to the recovery log.

reduce. To free up space from the database or recovery log, to allow you to delete a volume. Contrast with *extend*.

register. (1) To define a client node or administrator who can access the server. See *registration*. (2) To specify licenses that have been purchased for the server.

registration. The process of identifying a client node or administrator to the server.

restore. To copy information from its backup location to the active storage location for use. In Tivoli Storage Manager, you can restore the server database, storage pools, storage pool volumes, and users' backed-up files. The backup version in the storage pool is not affected by the restore operation. Contrast with *backup*.

retention. The amount of time, in days, that inactive backed-up or archived files are kept in the storage pool before they are deleted. Copy group attributes and default retention grace periods for the domain define retention.

retention period. On an MVS server, a device class attribute that specifies how long files are retained on sequential access media. When used, Tivoli Storage Manager passes this information to the MVS operating system to ensure that other tape management systems do not overwrite tape volumes that contain retained data.

retrieve. To copy archived information from the storage pool to the workstation for use. The archive version in the storage pool is not affected by the retrieve operation. Contrast with *archive*. See also *storage pool*.

rollback. To remove changes that were made to database files since the last commit point.

S

schedule. A database record that describes scheduled client operations or administrative commands. See *administrative command schedule* and *client schedule*.

scheduling mode. The method of interaction between a server and a client for running scheduled operations on the client. Tivoli Storage Manager supports two scheduling modes for client operations: *client-polling* and *server-prompted*.

scratch volume. A labeled volume that is either blank or contains no valid data, that is not currently defined to Tivoli Storage Manager, and that is available for use.

script. See *Tivoli Storage Manager command script*.

selective backup. The process of backing up selected files or directories from a client domain. incremental backup. See also *incremental backup*.

serialization. The process of handling files that are modified during backup or archive processing. See *static*, *dynamic*, *shared static*, and *shared dynamic*.

server migration. The process of moving data from one storage pool to the next storage pool defined in the hierarchy, based on the migration thresholds defined for the storage pools. See also *high migration threshold* and *low migration threshold*.

server options file. A file that contains settings that control various server operations. These settings, or options, affect such things as communications, devices, and performance.

server-prompted scheduling mode. A client/server communication technique where the server contacts the client when a scheduled operation needs to be done. Contrast with *client-polling scheduling mode*.

server storage. The primary and copy storage pools used by the server to store users' files: backup versions, archive copies, and files migrated from Tivoli Space Manager client nodes (space-managed files). See *primary storage pool*, *copy storage pool*, *storage pool volume*, and *volume*.

session resource usage. The amount of wait time, CPU time, and space used or retrieved during a client session.

shared dynamic. A value for serialization that specifies that a file must not be backed up or archived if it is being modified during the operation. Tivoli Storage Manager retries the backup or archive operation a number of times; if the file is being modified during each attempt, Tivoli Storage Manager will back up or archive the file on its last try. See also *serialization*. Contrast with *dynamic*, *shared static*, and *static*.

shared library. A library device that is shared among multiple Tivoli Storage Manager servers.

shared static. A value for serialization that specifies that a file must not be backed up or archived if it is being modified during the operation. Tivoli Storage Manager retries the backup or archive operation a number of times; if the file is being modified during each attempt, Tivoli Storage Manager will not back up or archive the file. See also *serialization*. Contrast with *dynamic*, *shared dynamic*, and *static*.

snapshot. See *database snapshot*.

source server. A server that can send data, in the form of *virtual volumes*, to another server. Contrast with *target server*.

space-managed file. A file that is migrated from a client node by the space manager client (HSM client). The space manager client recalls the file to the client node on demand.

space management. The process of keeping sufficient free storage space available on a client node by migrating files to server storage. The files are migrated based on criteria defined in management classes to which the files are bound, and the include-exclude list. Synonymous with *hierarchical storage management*. See also *migration*.

space manager client. The Tivoli Space Manager program that enables users to maintain free space on their workstations by migrating and recalling files to and from server storage. Also called *hierarchical storage management (HSM) client*.

startup window. A time period during which a schedule must be initiated.

static. A value for serialization that specifies that a file must not be backed up or archived if it is being modified during the operation. Tivoli Storage Manager does not retry the operation. See also *serialization*. Contrast with *dynamic*, *shared dynamic*, and *shared static*.

storage agent. A program that enables Tivoli Storage Manager to back up and restore client data directly to and from SAN-attached storage.

storage hierarchy. A logical ordering of primary storage pools, as defined by an administrator. The ordering is usually based on the speed and capacity of the devices that the storage pools use. In Tivoli Storage Manager, the storage hierarchy is defined by identifying the *next* storage pool in a storage pool definition. See *storage pool*.

storage pool. A named set of storage volumes that is the destination that the Tivoli Storage Manager server uses to store client data. The client data consists of backup versions, archive copies, and migrated files. You can back up a primary storage pool to a copy storage pool. See *primary storage pool* and *copy storage pool*.

storage pool volume. A volume that has been assigned to a storage pool. See *volume*, *copy storage pool*, and *primary storage pool*.

storage privilege class. A privilege class that allows an administrator to control how storage resources for the server are allocated and used, such as monitoring the database, recovery log, and server storage. Authority can be restricted to certain storage pools. See also *privilege class*.

stub file. A file that replaces the original file on a client node when the file is migrated from the client node to server storage by Tivoli Space Manager.

subscription. The method by which a managed server requests that it receive configuration information associated with a particular profile on a configuration manager. See *managed server*, *configuration manager*, and *profile*.

system privilege class. A privilege class that allows an administrator to issue all server commands. See also *privilege class*.

T

tape library. A term used to refer to a collection of drives and tape cartridges. The tape library may be an automated device that performs tape cartridge mounts and demounts without operator intervention.

tape volume prefix. A device class attribute that is the high-level-qualifier of the file name or the data set name in the standard tape label.

target server. A server that can receive data sent from another server. Contrast with *source server*. See also *virtual volumes*.

Tivoli Disaster Recovery Manager (DRM). A product that works with Tivoli Storage Manager to assist in preparing and later using a disaster recovery plan file for the Tivoli Storage Manager server.

Tivoli Storage Manager command script. A sequence of Tivoli Storage Manager administrative commands that are stored in the Tivoli Storage Manager server database. You can run the script from any interface to the server. The script can include substitution for command parameters and conditional logic.

U

UCS-2. An ISO/IEC 10646 encoding form, Universal Character Set coded in 2 octets. The Tivoli Storage Manager client on Windows NT and Windows 2000 uses the UCS-2 code page when the client is enabled for Unicode.

Unicode Standard. A universal character encoding standard that supports the interchange, processing, and display of text that is written in any of the languages of the modern world. It can also support many classical and historical texts and is continually being expanded. The Unicode Standard is compatible with ISO/IEC 10646. For more information, see http://www.unicode.org.

UTF-8. Unicode transformation format - 8. A byte-oriented encoding form specified by the Unicode Standard.

V

validate. To check a policy set for conditions that can cause problems if that policy set becomes the active policy set. For example, the validation process checks whether the policy set contains a default management class.

version. A backup copy of a file stored in server storage. The most recent backup copy of a file is the active version. Earlier copies of the same file are inactive versions. The number of versions retained by the server is determined by the copy group attributes in the management class.

virtual volume. An archive file on a *target server* that represents a sequential media volume to a *source server*.

volume. The basic unit of storage for the Tivoli Storage Manager database, recovery log, and storage pools. A volume can be an LVM logical volume, a standard file system file, a tape cartridge, or an optical cartridge. Each volume is identified by a unique volume identifier. See *database volume*, *scratch volume*, and *storage pool volume*.

volume history file. A file that contains information about: volumes used for database backups and database dumps; volumes used for export of administrator, node, policy, or server data; and sequential access storage pool volumes that have been added, reused, or deleted. The information is a copy of the same types of volume information in the Tivoli Storage Manager database.

Index

© Copyright IBM Corp. 1993, 2002

Program Number: 5698-ISM
5698-ISE

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC32-0768-00

Spine information:

Tivoli

Tivoli Storage Manager
for AIX

Administrator’s Guide

Version 5
Release 1