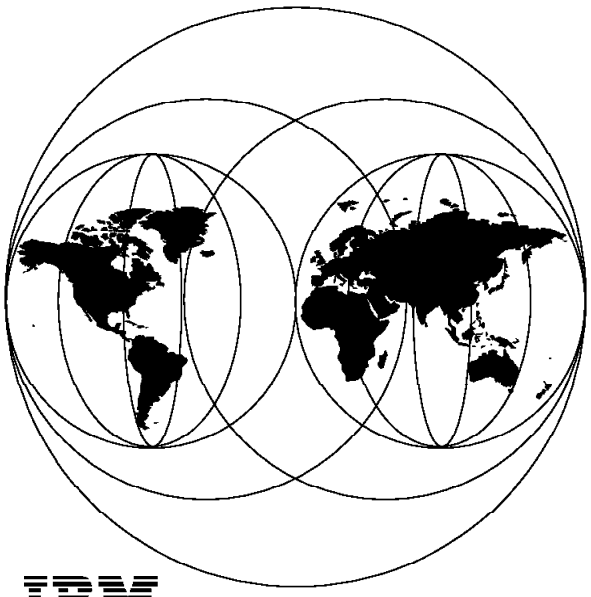
SG24-4878-00

ADSM Server for Windows NT Configuration and Recovery Examples

January 1997



IBM
International Technical Support Organization
San Jose Center

SG24-4878-00

International Technical Support Organization

ADSM Server for Windows NT
Configuration and Recovery Examples

January 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special Notices” on page 145.

First Edition (January 1997)

This edition applies to Version 2.1 of ADSM for Windows NT, Program Number 5639-A09.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

This redbook provides configuration and recovery examples for the ADSM Server for Windows NT. The examples cover installation, configuration tools, tape and library configuration, database recovery, storage pool recovery, and server disaster recovery.

This redbook is written for ADSM business partners, customers, consultants, and IBM personnel who want to implement the ADSM server on Windows NT.

The book is a tutorial for configuration and recovery of the server. The assumption is that the reader is familiar with ADSM on other platforms and wants to implement ADSM on Windows NT. Maximum benefit will be gained by using this book in conjunction with installing the ADSM server on Windows NT. No in-depth knowledge of Windows NT is required.

How This Redbook Is Organized

This redbook contains 163 pages. It is organized in two parts:

Part 1, “Server Configuration”

The first part covers installation and configuration of the ADSM for NT server.

- Chapter 1, “Installing ADSM for NT”

  This chapter covers the initial installation and verification of an ADSM for NT server. Its objective is to show how ADSM can be installed and used in 10 minutes.

- Chapter 2, “Server Configuration”

  This chapter introduces the various administrator interfaces available to an ADSM administrator on NT. It focuses on the new server utilities applications.

- Chapter 3, “ADSM Storage Devices”

  This chapter covers ADSM storage pool devices, the types of devices supported, removable media, and how they work together.

- Chapter 4, “Removable Media Configuration”

  This chapter gives examples of how to configure removable media to ADSM. It covers ADSM and NT device drivers, configuring devices to ADSM, preparing media, and defining these devices and media to ADSM storage pools.

Part 2, “Server Availability”

The second part builds on a basic configuration and looks in detail at the utilities provided for managing server availability.

- Chapter 5, “Database Availability”

  This chapter looks at ADSM server database availability. It covers database and recovery log mirroring, database backup, database recovery modes, and database recovery with examples of recovering a complete database and individual database volumes.

- Chapter 6, “Storage Pool Availability”

  This chapter looks at ADSM server storage pool availability. It covers storage pool backup and recovery with examples of recovering disk and tape volumes.

- Chapter 7, “Server Recovery Example”

  This chapter builds on the previous two chapters, showing how administrators can plan and prepare for server disaster recovery. It covers preparing the necessary configuration data and server scheduling and includes an example of recovering a server.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization San Jose Center.

Tim Mortimer is a Senior Systems Specialist at the International Technical Support Organization-San Jose Center. He writes extensively and teaches IBM classes worldwide on all areas of ADSM and distributed storage management. Before joining the ITSO one year ago, Tim worked for IBM in the United Kingdom as a storage management consultant. He has 20 years of IT experience, including 10 years of storage management experience covering a wide of mainframe and distributed systems.

Giulio John Tarella is a Certified ITAP Service Specialist in Italy. He has eight years of experience in storage and performance management on mainframes, servers, and workstations. He holds a degree in Seismic Structural Engineering from Politecnico di Milano. His areas of expertise include ADSM implementation and exploitation, MVS storage management, and MVS performance management. He has written extensively on DFSMS and ADSM implementation.

Achim Warnecke is a Senior Associate Programmer in Mainz, Germany working in the EMEA ADSM presales support area. He has two years of

experience in ADSM technical marketing support and works on Business Partner Enablement and customer support.

Michael Kindermann is a Systems Specialist at IBM Germany. He has two years of experience in software development, specializing in the Windows operating systems. Michael holds a degree in Computer Science. His areas of expertise include operating systems (Windows NT, OS/2, AIX), databases, implementation, programming, and support.

Thanks to the following people for their invaluable contributions to this project:

Dick Johnson
IBM SSD San Jose

Mike Collins
IBM SSD Tucson

and to

Maggie Cutler
ITSO-San Jose Center editor

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Part 1. Server Configuration

In part 1 of this book we cover the installation and configuration of the ADSM server for Windows NT, emphasizing removable media devices such as tape drives and tape libraries and their configuration. Topics covered are:

- Server installation
- Server configuration tools
- Server storage devices
- Removable media configuration examples

Chapter 1. Installing ADSM for NT

In this chapter we look at how the ADSM for NT server package is installed and the installation verified. Topics covered are:

- System requirements
- Installing ADSM components
- Verifying the installation

We discuss the ADSM for NT packages and their hardware, operating system, and file system requirements and explain how to install the ADSM for NT server package, including the server, local clients, and administrative clients. We conclude the chapter with a discussion of how you can quickly verify your installation by using the server and client on the same NT system and how to configure the communications protocols used.

1.1 System Requirements

ADSM for NT is available in two editions:

- ADSM for NT Single Server Edition enables the ADSM server and client to run on the same NT system. This edition provides base removable media device support for 4 mm, 8 mm, and QIC tape drives. Additional device support modules can be ordered for DLT drives, 4 mm libraries, 8 mm libraries, and DLT libraries. The single server edition does not support network-attached clients.
- ADSM for NT Network Server Edition supports the same removable media devices as the single server edition. Additionally it supports network-attached clients. Additional client authorizations can be ordered in quantities of 1, 5, 10, and 50 users. The full range of ADSM client platforms is supported.

The ADSM for NT Single Server and Network Server editions have the same operating system requirements:

- A Windows NT 3.51 or 4.0 workstation or server, with a minimum of 100 MB free disk storage (for a full installation) and 16 MB (32 MB is recommended) of memory
- An Intel based 486DX or Pentium processor

Non-Intel Hardware

The ADSM server and client do not support the other NT hardware platforms: Alpha, MIPS, and PowerPC.

The following communication protocols, standard with NT 3.51 and 4.0, are supported:

- Named pipes
- NetBIOS
- IPX/SPX
- TCP/IP

Named Pipes

Named pipes is supported only between a client and server running on the same system. Network-attached clients must use one of the other protocols.

ADSM supports both the file allocation table (FAT) file system and the NT file system (NTFS) on NT. The ADSM for NT package can be installed on a drive formatted as either file system. The choice of which file system to use depends on your own preferences. NTFS is a secure, reliable file system with good fault tolerance and recovery features. FAT does not have these availability features but is potentially faster than NTFS because it does not have the overhead associated with those features. FAT also allows other operating systems to access a FAT-formatted drive. This can be very useful in a situation where a system has to be booted with another operating system to recover a damaged system. In short, unless performance is the highest priority, NTFS is the logical choice for the ADSM server. Its availability features more than compensate for a slight performance reduction.

1.2 Installing ADSM Components

Installing the ADSM server for NT package is simple. The ADSM installation program provides a Windows InstallShield Wizard, which guides you through the setup process. We strongly recommend that you exit all Windows programs before running the ADSM installation program. Insert the ADSM for Windows NT CD into your CD-ROM drive and perform the following steps:

1. Start File Manager.
2. Double-click on the SETUP.BAT file located in the root directory of the ADSM CD.
3. Read the Welcome screen and click on **Next** to continue.

The ADSM installation program then gives you three choices for the installation (Figure 1 on page 5).

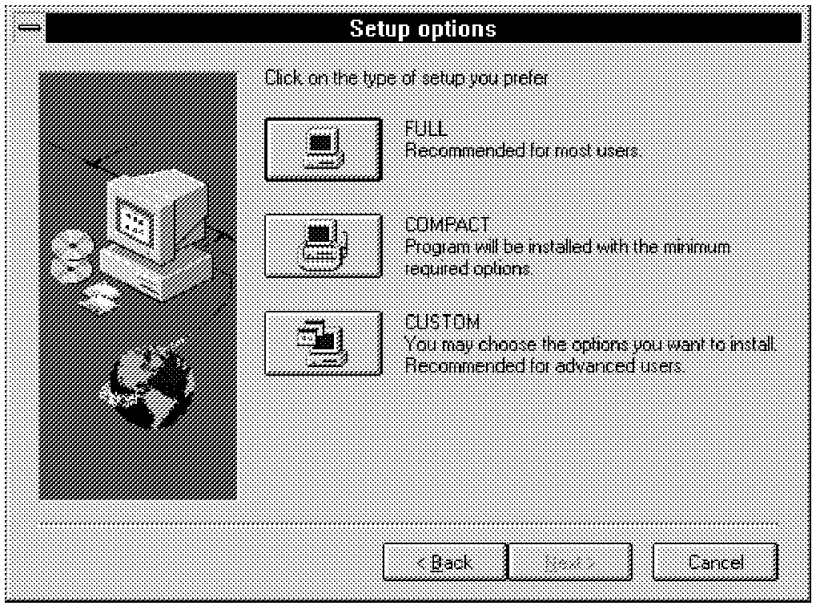


Figure 1. Setup Options Screen

- **FULL** installs all ADSM components: server, administrative clients, backup/archive clients, ADSM device drivers, utilities, and the online documentation.
- **COMPACT** installs all components of the FULL install with the exception of the online documentation.
- **CUSTOM** allows you to select which components are to be installed.

Once you have selected the installation option, the next screen displays the directory location where ADSM will be installed, C:\WIN32APP\IBM\ADSM by default. To install in a different directory, click on **Browse** and select a directory of your choice. Once you have chosen the directory, in the next screen you can choose a Windows Program Group name; ADSM for Windows NT is the default. You can choose an alternative name. Finally, click on **Next** to review or change these settings. When you are satisfied with the settings, click on **Next** to continue the installation process. A progress screen appears showing the files being installed. The following activities occur:

1. The selected ADSM components are installed.
2. A Windows Program Group for ADSM is created.
3. The ADSM server is initialized with the following resources:
   - A 4 MB disk storage pool volume: DATA1.DSM

- An 4 MB database volume: DB1.DSM
- An 8 MB recovery log volume: LOG1.DSM

The results of steps 1-3 are displayed in a window and logged to a file, INITSERV.LOG, in the server installation directory.

4. Initial configuration of the ADSM server is set up. This consists of defining the following to the server:

   - A client node named CLIENT
   - An administrator named ADMIN
   - The previously defined storage pool volume, DATA1.DSM, in the default backup storage pool
   - Sample daily and weekly incremental backup schedules

   The results of these actions are logged to a file, CFGSERV.LOG, in the server installation directory.

Finally, when this setup is finished, you are prompted to restart the system (Figure 2). System restart is required to activate the ADSMSCSI device driver and to include the ADSM server in NT's event logging.

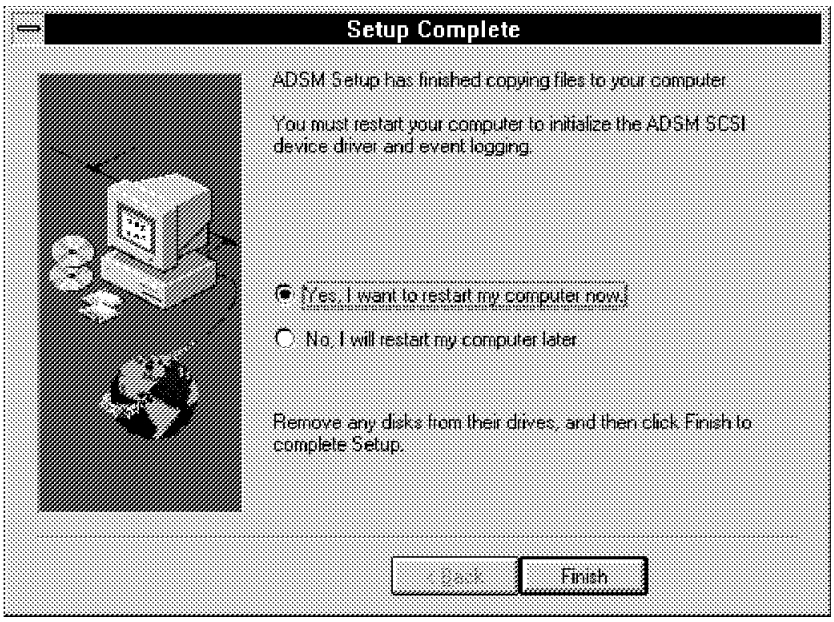


Figure 2. Setup Complete Screen

After restarting your NT system, you will see a new icon called *ADSM for Windows NT* or whatever name you chose. Double-click on that icon to open the ADSM for Windows NT window (Figure 3 on page 7). At this point, your ADSM installation is complete.

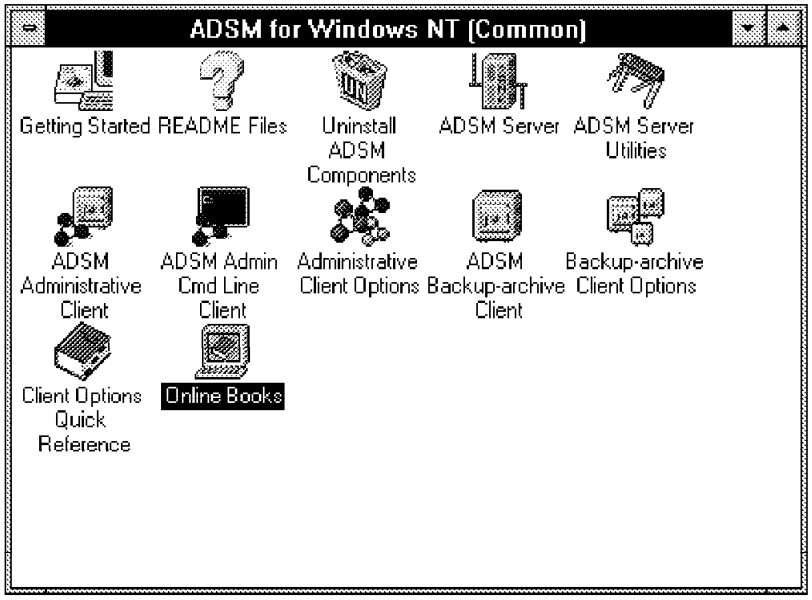


Figure 3. ADSM for Windows NT Program Window

On the ADSM for Windows NT window you will find icons for README Files, Client Options Quick Reference, Online Books, and Getting Started help, among others. With the full installation you also have the Bookmanager Library Reader, which gives you access to the ADSM client and server books.

NT 4.0

These examples are for NT 3.51. If you are using Windows 4.0, the same icons will be available but arranged as a cascading menu of ADSM icons from the **Start** button.

1.3 Verifying the Installation

To verify that the components have been installed correctly, you start the ADSM server and connect to it with a local client.

In this section we use the ADSM server and client installed on the same machine. Both can be run with the default configuration created during the installation process. Additional configuration of ADSM is required only if a protocol other than TCP/IP is required. The installation program creates a default backup-archive client node name (CLIENT) and password (CLIENT), which are used by the local backup-archive client.

1.3.1 Starting the Server

The ADSM server must be started before the client can connect to it. From the ADSM for Windows NT program folder, double-click on the **ADSM Server** icon. This action starts the server as a foreground session, displaying its initialization messages in an NT command window:

```
ANR0811I Inventory client file expiration started as process 1.
ANR8260I Named Pipes driver ready for connection with clients.
ANR0993I ADSM server initialization complete.
ANR8200I TCP/IP driver ready for connection with clients on port 1500.
ANR2560I Schedule manager started.
ANR2835I Server is licensed for 756 clients.
ANR2861I Server is licensed to support NETWORK connections.
ANR2843I Server is licensed to support UNIX clients.
ANR2844I Server is licensed to support desktop clients.
ANR2854I Server is licensed for device support module 2.

ADSM Server for Windows NT - Version 2, Release 1, Level 0.9/0.9

adsm>
```

This window displays the ADSM server messages during server initialization and for subsequent server activities. It is also the default server “console” where server commands can be entered if required. A series of messages will indicate that the various server resources and processes have been started. When successfully started, the server will display a command prompt, adsm>. At this point the ADSM server is ready for connections from a client.

Licensing

The NT command window shown above illustrates a server licensed for network-attached clients, UNIX clients, desktop clients, and full ADSM device support. Depending on the licensed package you are installing, your NT command window may differ from ours.

By default the ADSM server will start its TCP/IP and Named Pipe communication protocol drivers. The ADSM client installed as part of the installation process is configured to use TCP/IP. If TCP/IP is not configured on your NT system, the client must be reconfigured to use an alternative protocol. Alternative communication methods are covered in the next section. If TCP/IP is configured on your NT system, you can skip the next section and proceed to 1.3.3, “Using the Client” on page 10.

1.3.2 Alternative Communication Methods

The installation program configures the local ADSM client for TCP/IP connectivity. If you are not using TCP/IP, you must update the client configuration to use another protocol. Use the ADSM Client Configuration Wizards.

In the ADSM for Windows NT program window, double-click on the **ADSM Server Utilities** icon. The ADSM Server Utilities - Diagnostics window appears. A series of icons is arranged at the bottom of the window. Double-click on the **Client Configuration** icon to get to the Client Configuration Wizards (Figure 4).

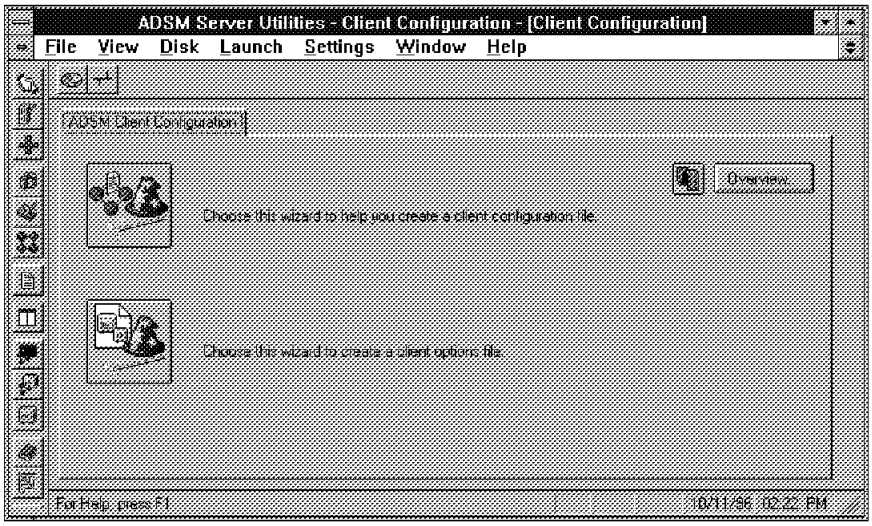


Figure 4. Client Configuration Wizards

There are two configuration wizards. The first wizard, which creates a client configuration file, is used for configuring network-attached clients. The second wizard, which creates a client options file, is used here.

As we discuss in 1.3.1, “Starting the Server” on page 8, the ADSM server by default listens for incoming client connections on TCP/IP and Named Pipes. For the purpose of verifying the ADSM installation by using the client on the same machine as the server, if TCP/IP is not configured, Named Pipes is the logical alternative. To reconfigure the local client for Named Pipes, use the client options file wizard to update the client options file:

1. Click on the client options file wizard icon to display an initial Client Configuration dialog box.

2. Click on **NEXT>>** to proceed to the next dialog box where you can select a communications protocol. By default the TCP/IP button is selected. Select the **Named Pipes** button and click on **NEXT>>**.
3. The Named Pipe Parameters dialog box is displayed with a default name in the named pipe box. This default name matches the configuration of the ADSM server and should be left as it is.
4. Clicking on **NEXT>>** again displays the Base Options dialog box. You can add an ADSM administrator name here if required. Ensure that the Append to base options file check box is not checked and then click on **NEXT>>**.
5. The Create an Options File dialog box is displayed. Here you can enter the client node name and options file name. For the node name enter CLIENT, which is the default node name registered on the server. For the options file name, enter C:\WIN32APP\IBM\ADSM\BACLIENT\DSM.OPT, which is the default directory and options file name created during the installation process.
6. Click on **FINISH** to complete the options file configuration (Figure 5).

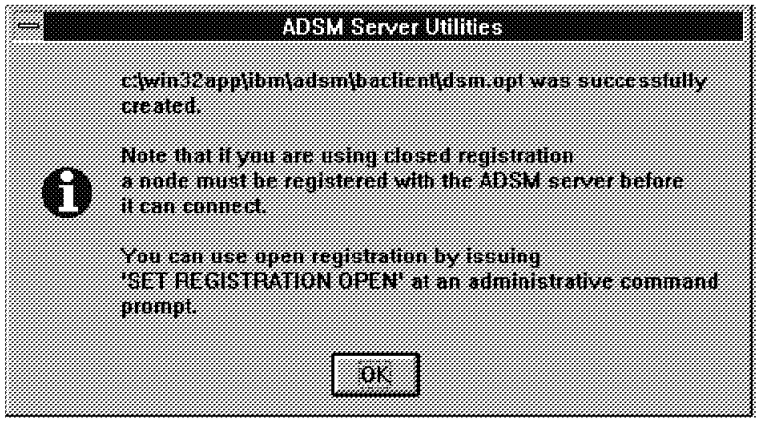


Figure 5. Client Configuration Complete Window

Once you have completed steps 1-5, you have reconfigured the local client to use Named Pipes as its communications protocol, and you can now use the client to verify the server operation.

1.3.3 Using the Client

The simplest way of verifying a successful installation, using the client, is to connect to the server and back up some data. Open the ADSM for Windows NT program window and double-click on the **ADSM Backup-archive Client** icon. This action starts the backup-archive GUI client, which will start a

session with the server and display a password prompt (Figure 6 on page 11).

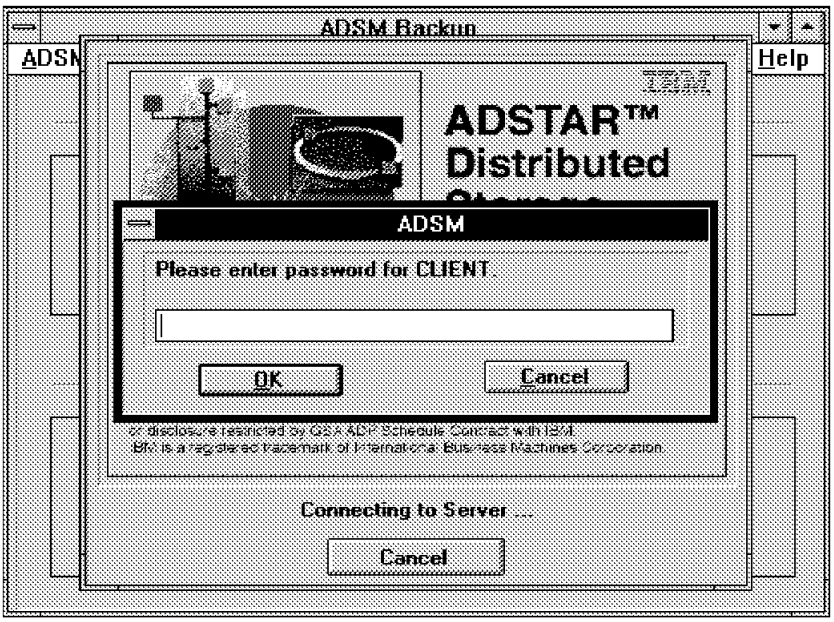


Figure 6. Client Password Prompt

The client has connected to the server using the configuration and node name parameter defined in the client options file. The server, when installed, is configured with a default client node name of CLIENT and a password of CLIENT. Enter this password and click on **OK** to complete the establishment of the client session.

At this point you have verified that the ADSM server and client can communicate through the chosen protocol. If you entered the password correctly, the client backup window will be displayed showing the drives on the local system that can be backed up to the ADSM server (Figure 7 on page 12).

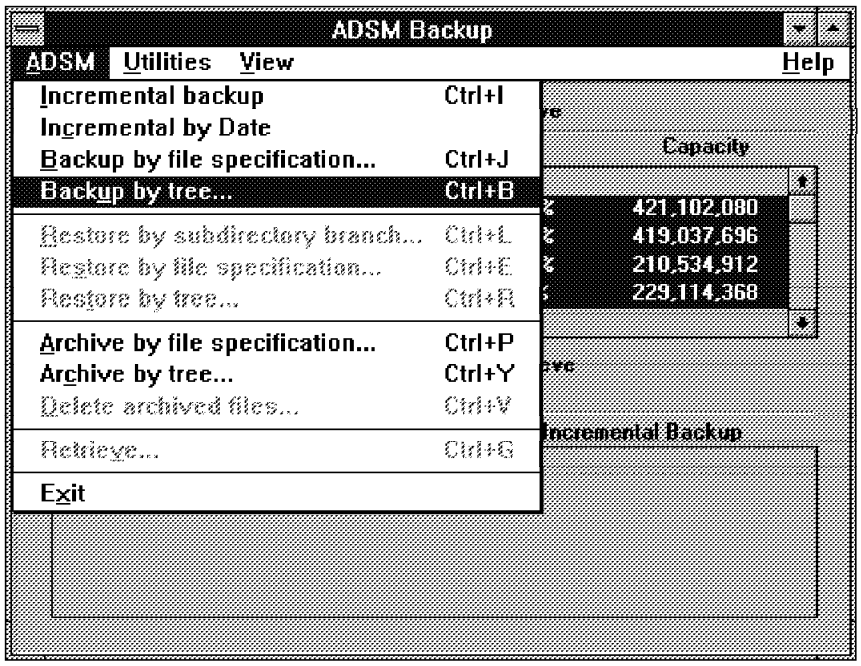


Figure 7. Client Backup Window

You can now back up data to the ADSM server. Assuming that no additional server configuration has taken place, the default server storage pool capacity is 4 MB. If you try to back up more than 4 MB, the server will run out of space, and the backup will terminate. To validate that the server is functioning correctly, perform a selective backup of a small number of files.

On the ADSM Backup window menu bar, click on **ADSM** and then select **Backup by tree**. This action displays, in a tree structure, the directories on the local system drives. Select a directory and files to be backed up to the server. When the backup is complete, an ADSM pop-up message appears indicating that the backup has completed successfully (Figure 8 on page 13).

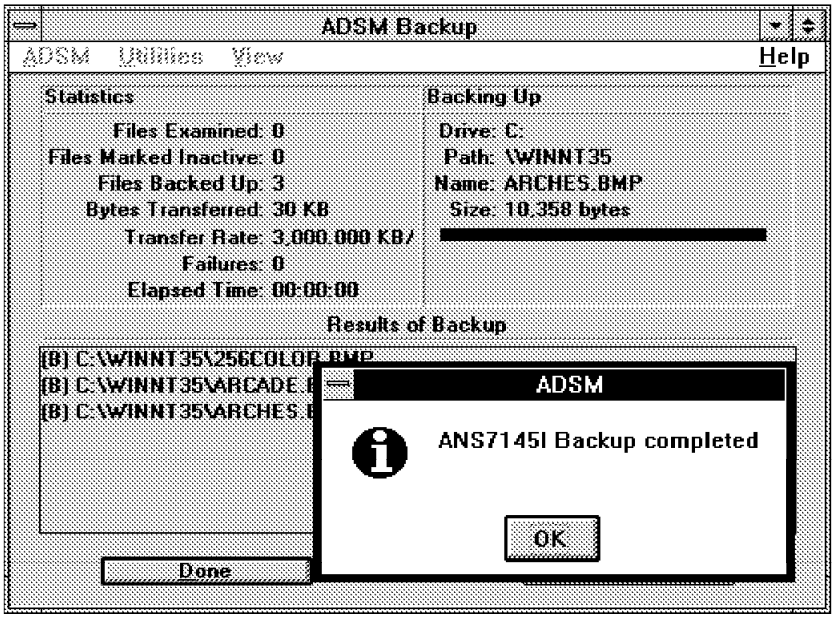


Figure 8. Backup Completed Message

This successful backup indicates that:

- ADSM is installed correctly.
- The server and client, installed on the same machine, can communicate through the chosen communications protocol.
- The client can back up data to the default disk storage pool on the server.

You have now successfully installed ADSM and verified that it is a working configuration.

Chapter 2. Server Configuration

In this chapter we consider the new tools available for configuring and administering an ADSM server running on NT. We assume that you are familiar with ADSM and its administrative clients.

2.1 Configuration Tools

You can configure the ADSM server using three different interfaces: the server console, administrative clients, and new server utilities.

2.1.1 Server Console

When started, the ADSM server directs it output messages to a console. If the server is started as a foreground session (by clicking on the **ADSM Server** icon) the server console window is displayed.

```
ANR0811I Inventory client file expiration started as process 1.
ANR8260I Named Pipes driver ready for connection with clients.
ANR0993I ADSM server initialization complete.
ANR8200I TCP/IP driver ready for connection with clients on port 1500.
ANR2560I Schedule manager started.
ANR2835I Server is licensed for 756 clients.
ANR2861I Server is licensed to support NETWORK connections.
ANR2843I Server is licensed to support UNIX clients.
ANR2844I Server is licensed to support desktop clients.
ANR2854I Server is licensed for device support module 2.

ADSM Server for Windows NT - Version 2, Release 1, Level 0.9/0.9

adsm>
```

You can use this console to monitor server activity and enter administrator commands. If the server is run as an NT service, this console window does not exist. With this configuration, certain server messages such as startup and critical error messages are logged in the NT application event log. Figure 9 on page 16 shows an example of a server error message during startup. In this case, a storage pool disk volume was unavailable when the server started. If the ADSM server is run as an NT service, a console interface is not available, so the server must be administered with an administrative client.

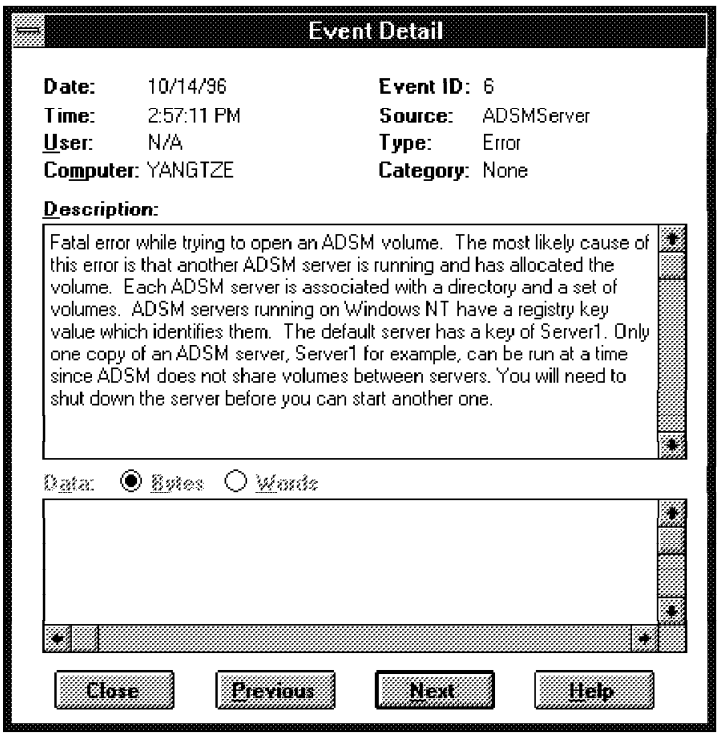


Figure 9. NT Application Event Log Detail. Server Startup Error

2.1.2 Administrative Clients

The administrative clients enable you to monitor and administer server activities, define management policies for client files, and set up schedules to provide services at regular intervals. You can use these clients to manage the ADSM server from a remote workstation rather than from the server console.

There are two administrative client interfaces, the graphical user interface (GUI) (Figure 10 on page 17) and the command line interface. Most of the commonly used ADSM functions are available from the GUI, and all functions are available from the command line interface.

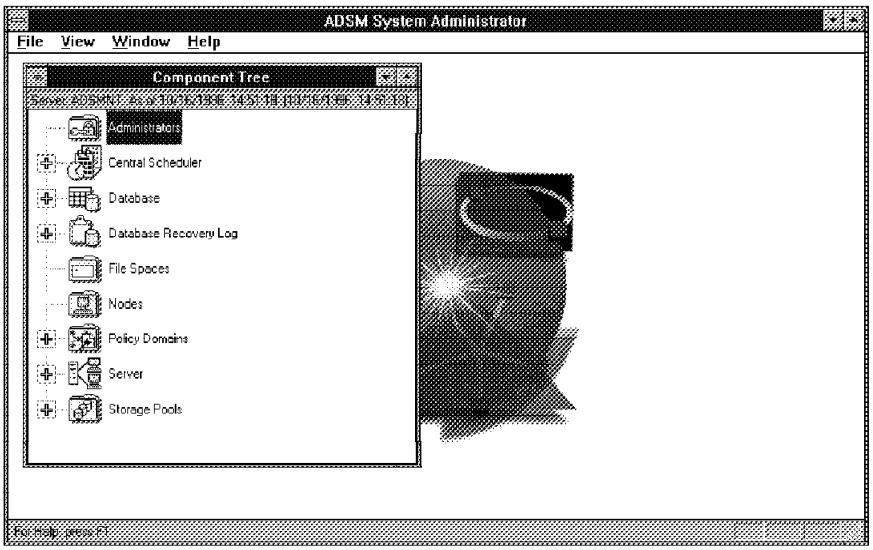


Figure 10. GUI Administrative Client Window

When the server is run as an NT service, it does not have a console interface for monitoring server activity. It this situation the administrative command line client is very useful. In addition to the normal interactive mode, where commands are entered, it can be run as a console monitor or a mount monitor.

Running as a console monitor, it displays all server messages that would typically be displayed on the server console. Running as a mount monitor, it displays mount messages for removable media volumes. These two modes are purely for monitoring; commands cannot be entered when the administrative client is started this way.

With the NT server, you can start the administrative clients by clicking on the appropriate icons in the ADSM program group. Alternatively, you can use the ADSM Server Utilities program to launch these clients in their various operating modes.

2.1.3 Server Utilities

The server console and administrative clients on NT provide the same interface for administration as on other ADSM platforms. Existing administrative clients on other platforms (AIX and OS/2, for example) can be used to administer the server running on NT. ADSM on NT provides a new interface for administrators—the Server Utilities application—which is installed during server installation and provides a single point for all NT

server administration. The Server Utilities provide additional configuration applications and can be used to “launch” an ADSM server running as a foreground application and any of the clients (Figure 11 on page 18).

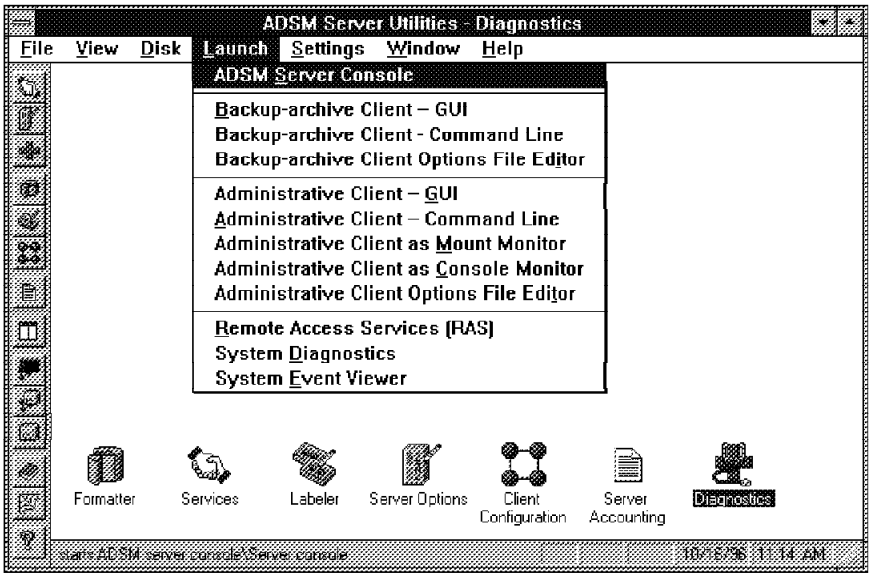


Figure 11. ADSM Server Utilities: Launch Menu

The configuration applications, shown as icons in Figure 11, provide a variety of functions: configuring NT services and resources, configuring ADSM server and client options, and formatting ADSM media.

2.2 Server Configuration

In this section we provide examples of using the ADSM Server Utilities to configure and administer the ADSM NT server.

2.2.1 Server Options

The basic server parameters are held in the server options file, DSMSERV.OPT. As with the other server platforms, DSMSERV.OPT is a text file that can be edited to update the various options. The server options application provides a GUI for updating these options. Click on the **Server Options** icon to get to the Server Options dialog, where a list of installed servers and their options files is displayed (Figure 12 on page 19).

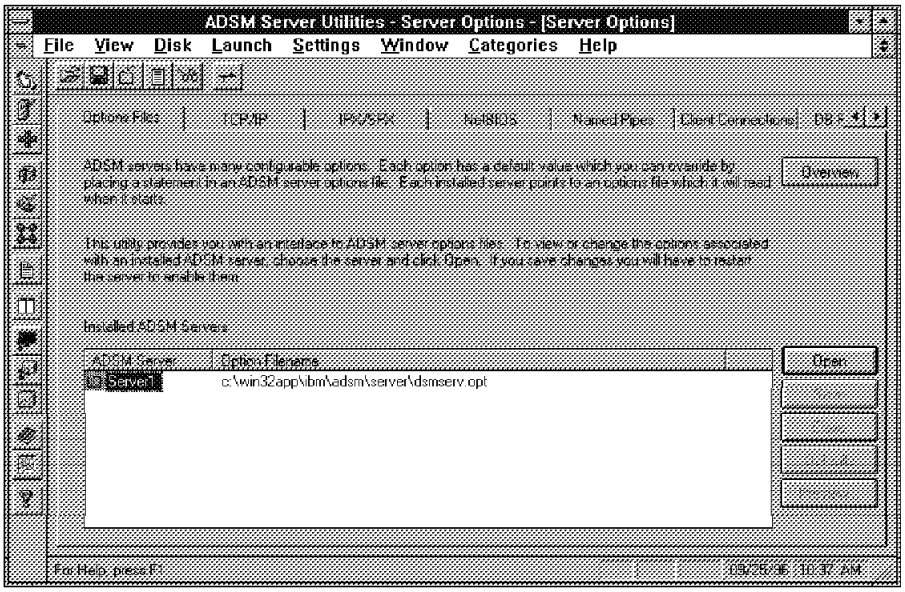


Figure 12. ADSM Server Utilities - Server Options

After you have selected the server to be configured, you can get the various server configuration options by selecting the appropriate tabbed page for the required server option. For example, to enable IPX/SPX for the server:

1. Select the IPX/SPX page.
2. Check the Enable IPX/SPX check box.
3. Go back to the first page (options files).
4. Click on the **Save** button.

The technique is the same for the other server options in DSMSERV.OPT. The application does not activate the updated options automatically; it just adds them to the options file. For the updated options to take effect, you must stop and restart the server.

2.2.2 NT Services

Three ADSM components can be run as NT services:

- The ADSM server
- The ADSM SCSI device driver
- The ADSM client scheduler

To configure these components, click on the **Services icon** in Server Utilities (see Figure 11 on page 18). Three tabbed pages are displayed: ADSM Server, Device Driver, and Scheduler for the local machine on which you are working (Figure 13 on page 20). Use the **Connect** button to connect to and work with these services on other NT machines.

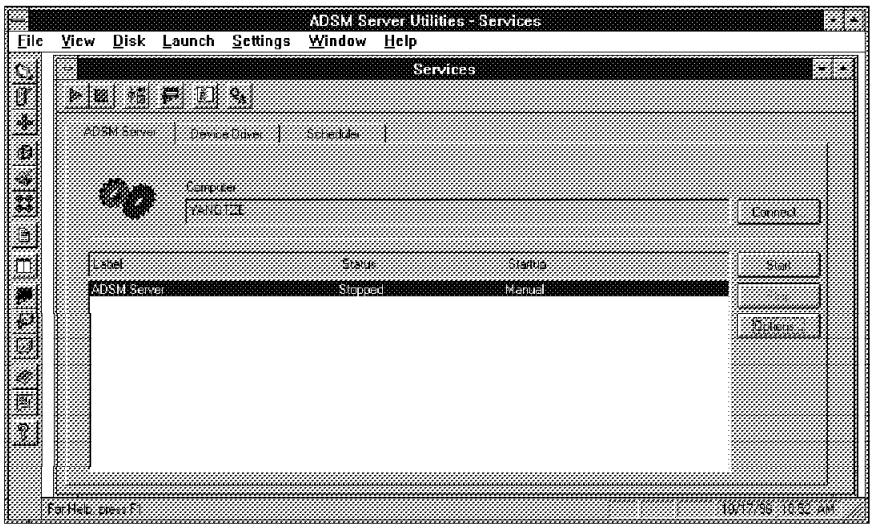


Figure 13. ADSM Server Utilities - Services

The ADSM installation process automatically creates these NT services. It does not start the services, however. They must be manually started by clicking on the **Start** button and stopped by clicking on the **Stop** button in the Services window.

You can also configure the services from the Services window. To configure the ADSM Server service to start automatically, click on the **Options** button. The ADSM Server Service Options dialog box will display where you can set the service to start automatically (Figure 14 on page 21). When you configure the server to start automatically, the server will start when the NT system is first booted. You do not have to log on locally to start the server.

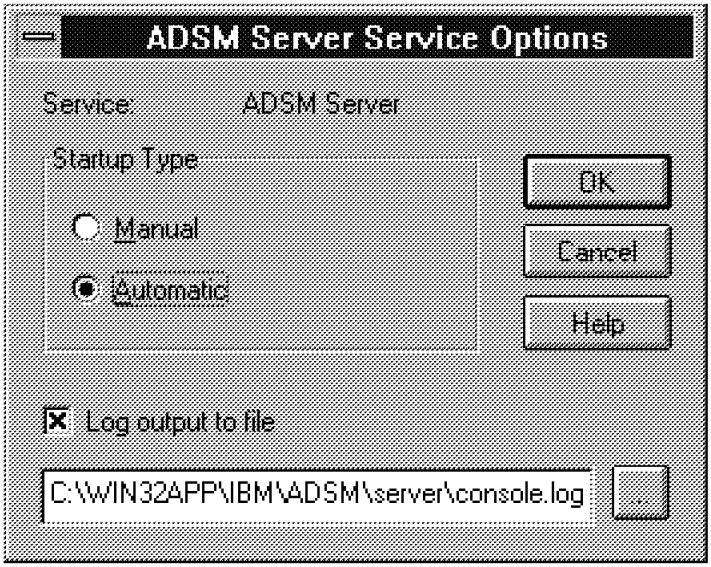


Figure 14. Setting ADSM Server Service to Automatic

Setting the ADSM Server service to start automatically is sensible, but the server must be administered from an administrative client. The server logs certain error messages to the NT event logger, but there is no server console to monitor activity. When configuring the ADSM Server as a service, there is an option to log output to a file, CONSOLE.LOG, in the server directory. This could be useful for diagnostic purposes in situations where the server will not start because of some problem and you cannot connect to it with an administrative client.

Note

This console output is not written to the CONSOLE.LOG file until the server is stopped.

2.2.3 Diagnostics

The server diagnostics provide an overview of the detected storage devices and the ADSM components installed on the NT system. The Device Detector is of particular use in configuring the server. It displays all of the SCSI-attached devices on the NT system and provides the majority of the information required to configure tape libraries and drives to ADSM (Figure 15 on page 22). By default the Device Detector displays devices on the local system. Click on the **Connect** button to browse other NT systems on the network as well.

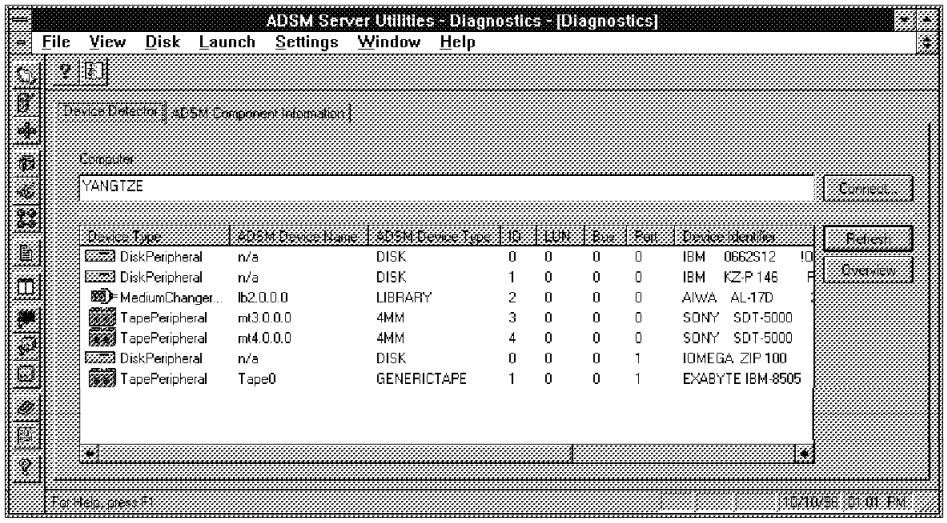


Figure 15. ADSM Server Utilities - Diagnostics

This Device Detector displays:

- Device type: disk, library, or tape
- ADSM device name, for library and tape devices, used during device configuration
- Recommended ADSM device type, used in configuring device classes
- The SCSI ID, LUN, Bus, and Port number for the device
- Device description

Use of this information is discussed in more detail in 4.1, “NT Device Drivers” on page 45.

2.2.4 Server Accounting

The Server Accounting application displays the ADSM server accounting log, DSMACCNT.LOG. For each client session, such information as date, time, and duration of the session is shown. The application does not format or report on the log entries; it just displays the entries (Figure 16 on page 23).

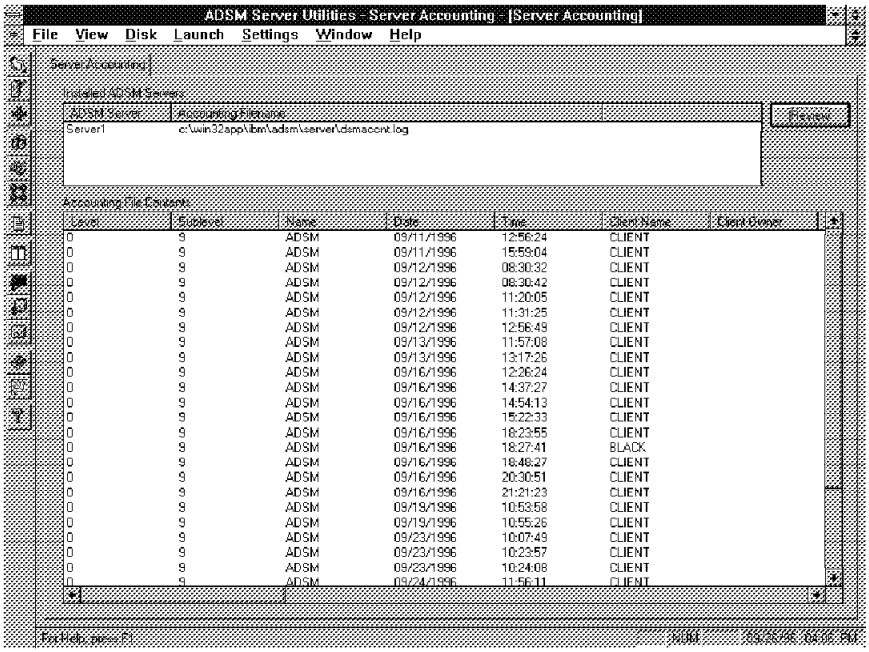


Figure 16. ADSM Server Utilities - Server Accounting

2.2.5 Formatting Server Disk Volumes

During ADSM server installation, space is allocated for the database, recovery log, and an initial backup storage pool volume on the server. This initial capacity is sufficient to validate that the server and local client work properly. To prepare the server for use with other clients, additional database, recovery log, and storage pool volume capacity is required. With previous ADSM server platforms, providing additional capacity involved:

1. Using the DSMLABEL utility to format the volumes
2. Defining the volumes to the server, either at the console or with an administrative client

This method is still available with the NT server; however, new formatting wizards have been provided. Click on the **Formatter** icon in the ADSM Server Utilities-Diagnostics window to get to the ADSM Server Utilities-Formatter window (Figure 17 on page 24).

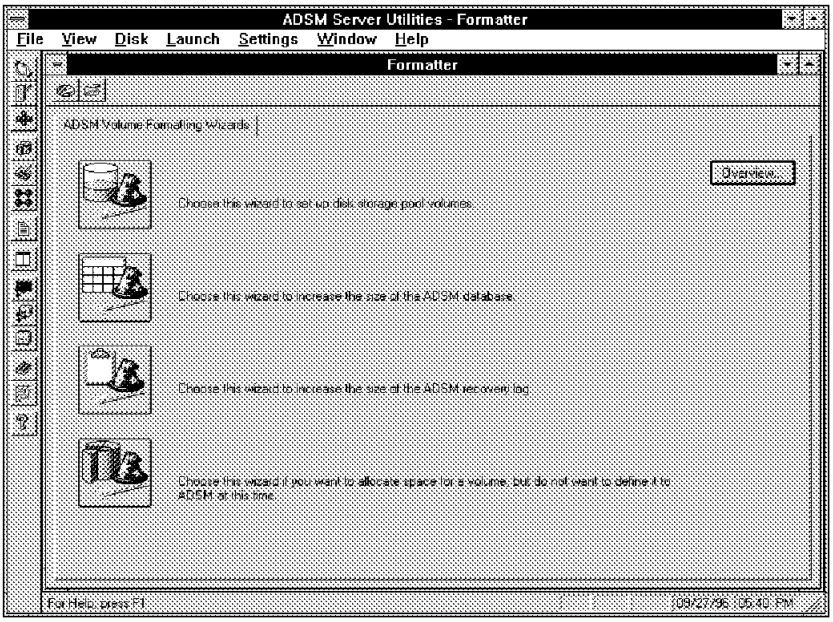


Figure 17. ADSM Server Utilities - Formatter

There are four formatting wizards, each of which performs a separate function:

1. Setting up disk storage pool volumes
2. Increasing the size of the database
3. Increasing the size of the recovery log
4. Formatting a volume, without defining it to the server

These wizards differ from the other server utilities in that they communicate with the running server to define the resources. The other server utilities are a combination of offline ADSM utilities and native NT tasks. When the wizards are invoked, they format the specified resource and define it to the server, using an administrative client session. Below we provide examples of using wizards to define storage pool volumes and database volumes.

2.2.5.1 Defining Storage Pool Volumes

To increase the size of a disk storage pool by adding additional volumes, click on the top wizard in the Formatter window. This action starts a dialog that enables you to select the storage pool to be configured and the administrator ID and password to be used. The next dialog asks you for an estimate of the amount of client data to be managed. After you have entered a value, the next dialog provides recommendations for the suggested

storage pool and disk volume size on the basis of the quantity of managed data you previously entered (Figure 18 on page 25).

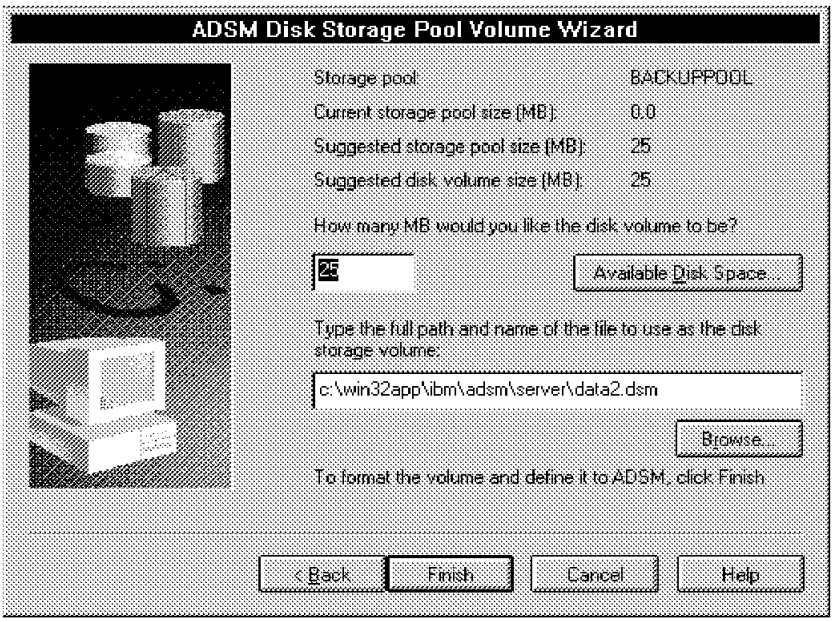


Figure 18. Disk Storage Pool Volume Wizard

At this stage you can change the size and location of the volume and check that there is sufficient available disk space for the volume to be created. When satisfied, click on **Finish** to format the new volume and define it to the server, using the administrator ID and password previously entered.

2.2.5.2 Defining Database Volumes

To define an additional database volume, click on the second wizard in the Formatter window (Figure 17 on page 24). This action starts a dialog asking you how much client data you intend to manage on the server. Using the value you give, the ADSM database volume wizard calculates a target database size and displays another dialog box with the current database size, recommended database volume size, and details of a new database volume (Figure 19 on page 26).

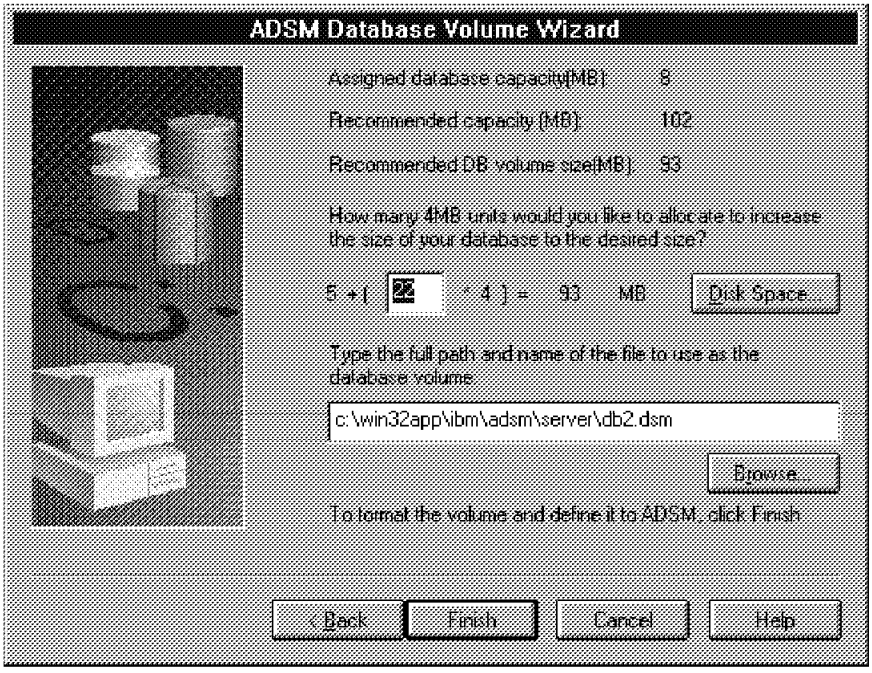


Figure 19. Database Volume Wizard

You can change the suggested size and name of the volume as well as check on the available disk space. When satisfied, click on **Finish** to format the volume, define it to the server, and extend the database to use the new capacity. These actions are done through an administrative client session.

2.2.6 Tape Labeling

You can use the ADSM labeling wizard to label media on manual or autochanger devices. For example, to label a tape on a manual tape drive:

1. Choose the wizard to label media on manual devices.
2. Read the information and click on **Next**.
3. In the ADSM Manual Device Media Labeling Wizard Window, enter the label name, check Overwrite existing label, select a manual drive, and click on **Apply** (Figure 20 on page 27).

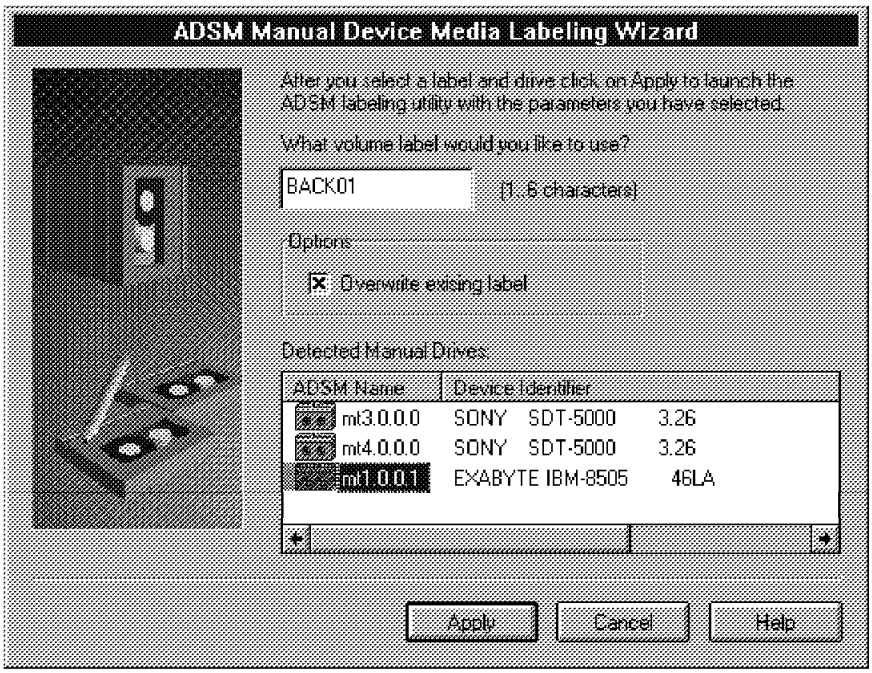


Figure 20. Manual Device Media Labeling Wizard Window

4. Click on **Finish** to generate a command line that labels your media on the device (be sure to have some media in the device).

Defining removable media devices and labeling the media are covered in more detail in Chapter 4, “Removable Media Configuration” on page 45.

2.3 Client Configuration

In this section we look at the client configuration tools and explain how to create a client configuration file and configure alternative communication protocols. In the ADSM Server Utilities-Diagnostics window, click on the **Client Configuration** icon to display the client configuration wizards (Figure 21 on page 28).

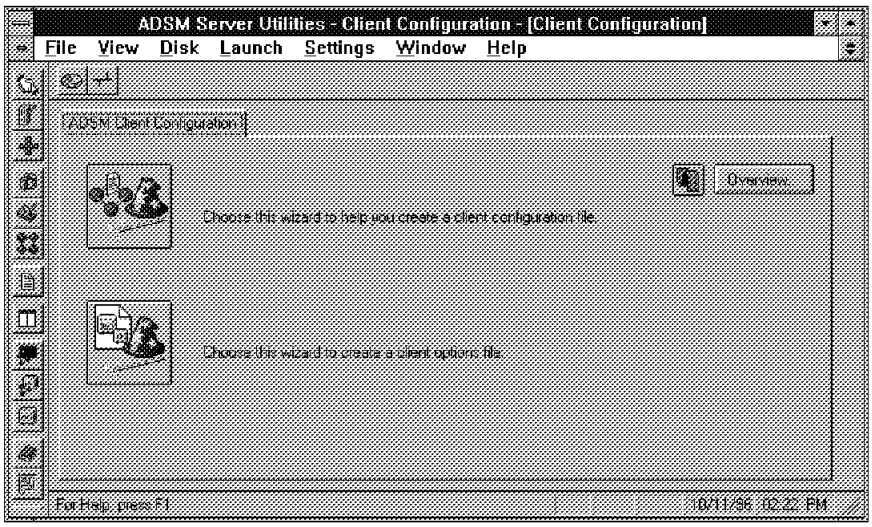


Figure 21. ADSM Server Utilities - Client Configuration

The client options file wizard is an options file editor used to create or update options files on the ADSM server file system. A locally installed client or network-attached clients can use these options files. Use of the client options file wizard is discussed in 1.3.2, “Alternative Communication Methods” on page 9. The client configuration file wizard is a new tool for creating an options file “package.” When this package is executed on a client workstation, an options file is automatically created.

2.3.1 Client Configuration Files

Every client has an options file that defines the connectivity options, client node name, and other ADSM processing options for that client. The file, typically DSM.OPT, can be configured by editing it on the local workstation where the client is run.

The client configuration file wizard enables you to create a configuration file that creates an options file on the workstation where it was executed. Clicking on the wizard starts a dialog where you can define:

- The connectivity protocol to be used
- The ADSM server address for the selected protocol
- The name of a base options file to be used

This wizard creates the appropriate connectivity parameters for a client to connect to the server. It also enables you to append the contents of a base options file to the configuration you are creating. This base options file can be created with other client options that are common to a number of client

workstations, such as include/exclude lists, compression, and language format options.

Three steps are involved in creating and executing a client configuration file (Figure 22):

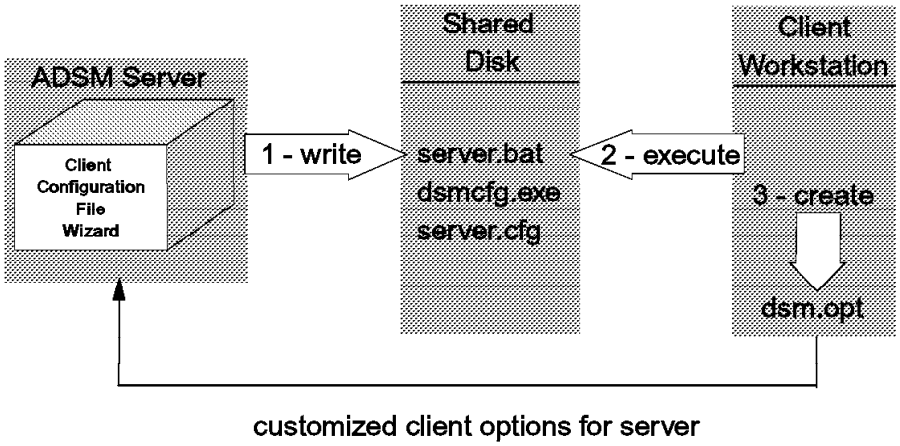


Figure 22. Creating a Client Configuration File

1. The client configuration file wizard is run on the ADSM server, specifying the connectivity options, base options file, and the name and location where the configuration file is to be stored. (The file should be stored on a shared disk, accessible by the client workstations on the network.) The wizard creates three files in the specified directory:
   - SERVER.BAT, used to invoke DSMCFG.EXE
   - DSMCFG.EXE, the client configuration utility
   - SERVER.CFG, contains the chosen option file parameters, used by DSMCFG.EXE to create the client options file

   Multiple configuration files can be created with different options by specifying a different configuration name, such as SERVER2.CFG, SERVER3.CFG, and so forth.
2. A client workstation elsewhere on the network accesses the shared directory and executes the configuration batch file, in this example, SERVER.BAT.
3. Executing SERVER.BAT on the workstation invokes DSMCFG.EXE, which uses the configuration file, SERVER.CFG, to set the appropriate options. The dialog on the workstation (Figure 23 on page 30) prompts you for the client node name to be added, by default the NT machine name, and

shows the directory path where the options file will be created. Click on **OK** to create the options file on the client workstation.

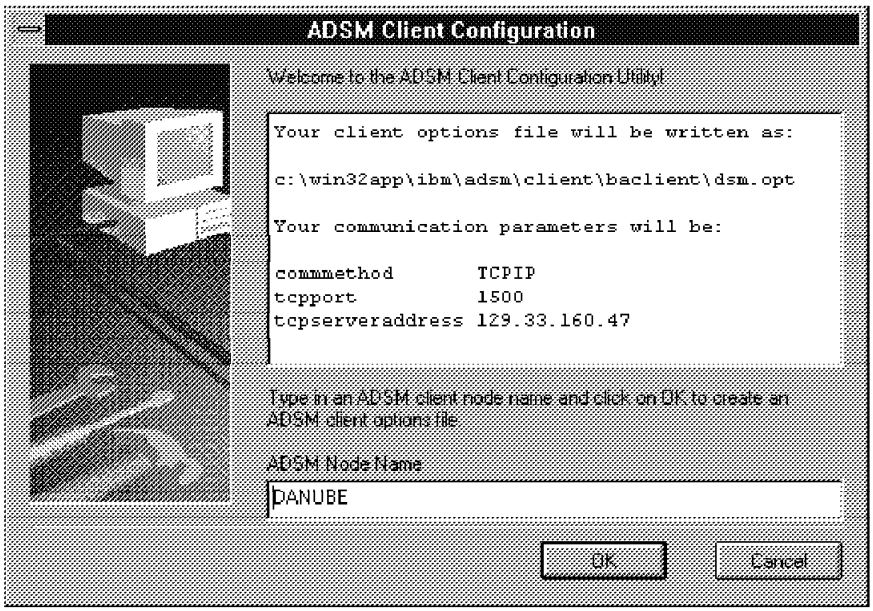


Figure 23. Executing Client Configuration File

The client configuration file wizard does not install the client code. The code must be installed locally or made available by creating a network shared directory. The wizard does, however, simplify the centralized configuration of clients.

Win32 Only!

With the ADSM NT server the configuration utility, DSMCFG.EXE, is a Windows 32 bit application. Therefore, configuration files created with the client configuration file wizard can be executed only on Windows NT and Windows 95 client workstations.

2.3.2 Configuring Communication Protocols

Configuring clients to use existing network protocols is straightforward. Adding protocols and configuring them with ADSM servers and clients are more complex operations. In this section we assume that the default Novell NetWare protocol, IPX/SPX, is not currently configured and look at how it is configured on NT and the ADSM server (which is particularly important if NetWare ADSM clients are to be implemented).

You can easily determine which protocols are active by starting the client configuration wizard and clicking on the **Network** icon in the action bar (Figure 24 on page 31).

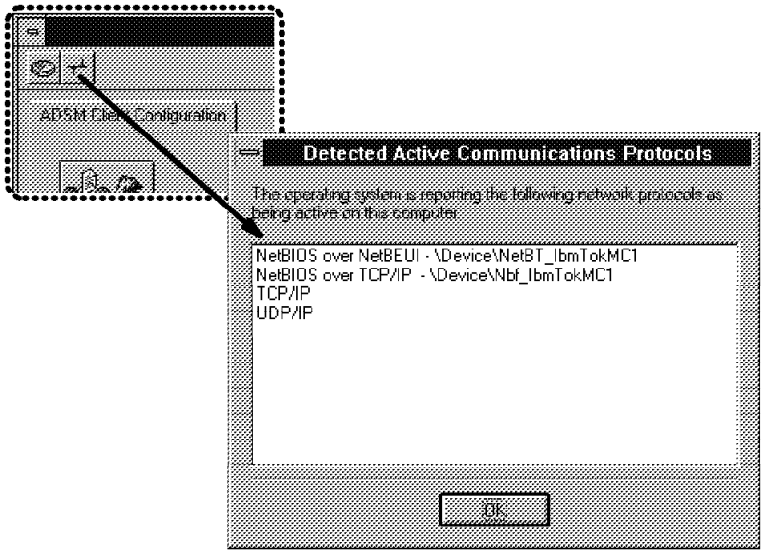


Figure 24. Determining Active Protocols

To add another protocol, open the NT control panel application and double-click on the network applet. The installed network adapter cards and software (Figure 25) are displayed.

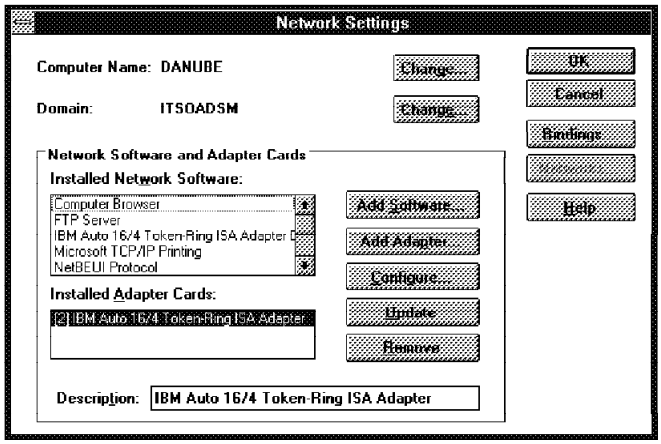


Figure 25. NT Network Settings

To add IPX/SPX, click on **Add Software...** In the Add Network Software dialog box (Figure 26 on page 32) you can select the network software to be added.

Figure 26. Selecting a New Network Protocol

Select NWLink IPX/SPX Compatible Transport and click on **Continue**. You are prompted for the NT installation CD if the required drivers are not already installed. When the installation of the required network drivers is complete, you are returned to the Network Settings dialog box (Figure 25 on page 31). Having installed the protocol, you must now “bind” it to the network adapter card. Click on **Bindings** to display the protocols that will be bound to the adapter card (Figure 27).

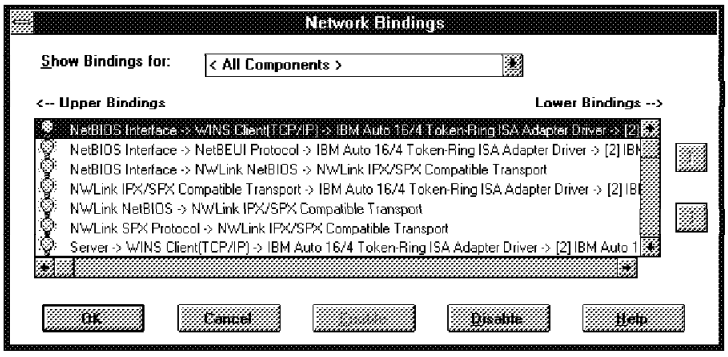


Figure 27. Network Bindings

Click on **OK** to complete the setup of the new protocol. Before the protocol can be activated, you must stop and restart NT.

After adding IPX/SPX and restarting NT, the protocol should be available for use by the ADSM server. Open the ADSM Server Utilities-Diagnostics window and click on the **Server Options** icon. You can verify here that the new protocol is active by again clicking on the **Network** icon in the action bar (Figure 24 on page 31). The newly added IPX/SPX protocol will be listed.

Next open the options file for the server. Go to the IPX/SPX page and check Enable IPX/SPX for ADSM client/server communications to display the IPX/SPX socket number and buffer size to be entered in DSMSERV.OPT.

Also on the IPX/SPX page is a **Client Settings** button. Click on it to display the appropriate ADSM server IPX/SPX address that can be used in a client options file (Figure 28).

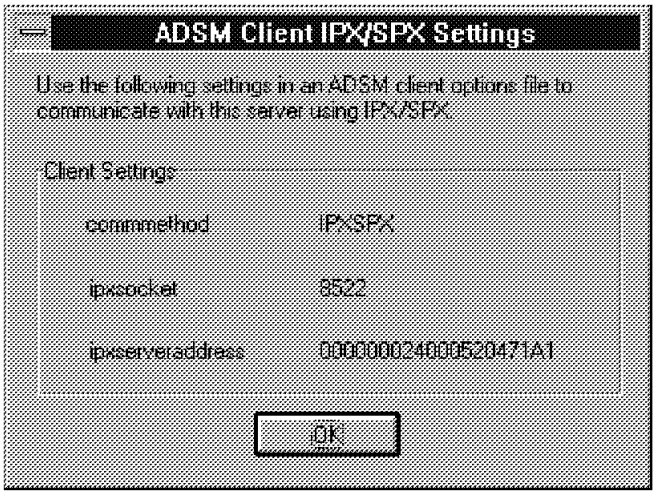


Figure 28. Server IPX/SPX Address

You can add these values by using the client configuration wizards or editing the client options file for those platforms that do not currently support the wizards.

Having enabled the new protocol in the server options file, DSMSERV.OPT, this file must be saved and the server restarted before they are available for client connections.

Chapter 3. ADSM Storage Devices

ADSM stores client backup and archive data in storage pools. Storage pools consist of one of more volumes. These volumes can be either logical volumes created on disk or physical removable media volumes that must be mounted before they can be used. In this chapter we briefly discuss both ADSM hardware devices and removable media, which are fully covered in the *ADSM for Windows NT Administrator's Guide*. Our brief review of these concepts here will facilitate the task of configuring storage pools with removable storage devices.

3.1 Hardware Devices

ADSM represents storage devices with definitions defined by the administrator. These definitions are device class, library, and drive. Each definition contains information that ADSM required to manage the device and its media. All devices require as a minimum a device class. The device class represents the type of device, its capacity, and the way in which the device accesses its media. For sequential media such as tape, the device class contains information about the format and capacity of the tape and a reference to the library in which it is used. The library and drive definitions are required only for specific kinds of devices such as removable media.

3.1.1 Disk Devices

ADSM storage pool volumes on disk can be defined for random access or as sequential files.

3.1.1.1 Random Access

ADSM disk devices normally use the predefined disk device class (Figure 29), which defines the disk device as a random access, fixed disk.

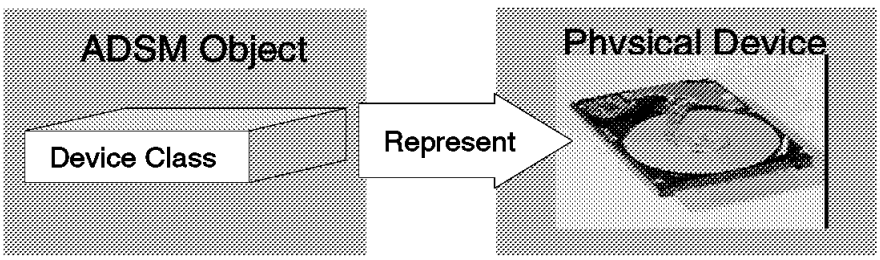


Figure 29. Disk Device Class

Server volumes created with the disk device class are treated as nonremovable. When the server is started, the volumes are varied online automatically and must always be available. The server assumes that any data stored with a disk device class can always be accessed without having to “mount” the device in some manner.

3.1.1.2 Sequential Disk Files

Administrators can define logical volumes on server disks that ADSM handles in the same way it handles other sequential access volumes such as tape. These volumes use a device class with a device type of *file*. This type of device class contains a reference to the drive and directory that will contain the files that ADSM will create.

ADSM writes to files that are located on drives such as the C: or D: drive. These drives can be local or remote to the ADSM server system and can reside on permanently mounted or removable media such as diskettes or other removable media devices. ADSM does not handle mount operations for files created in this manner on removable drives. It expects them to be available on demand.

A device class of this type could be created for use with a Network File System (NFS) mounted drive. This could be useful for creating copies of the ADSM server data that do not reside on the server, to be used for server recovery in the event of a server disk failure. For example, you might choose to back up the server database, using a device class pointing to an NFS mounted drive.

3.1.2 Removable Media Devices

Removable media devices are defined as libraries to ADSM. A library is a collection of one or more drives and, optionally, robotic autochanger devices used to mount the media in the drives. Every device that can use removable media must be associated with a library. ADSM uses the library definition to determine on which drive it can mount a particular type of media.

The device class is the basis for defining a library to ADSM. It defines the type of media, its capacity, how it is accessed, and the library in which the drives exist. This combination of device class, library definitions, and one or more drive definitions represents a library to ADSM (Figure 30 on page 37).

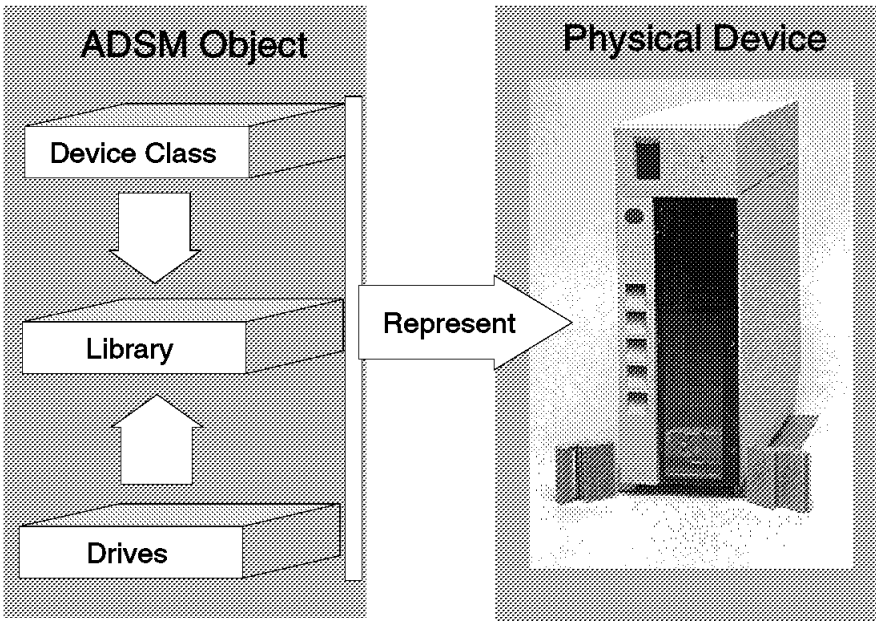


Figure 30. Library Representation

ADSM supports a number of tape device types for use in defining device classes: 4 mm, 8 mm, QIC, DLT, and GENERICTAPE. When defining a device class for use with a library, the appropriate device type is specified with its format. If an Exabyte 8 mm library were being defined, a device class would be created using a device type of 8 mm and the appropriate format, for example 8500C for a drive that records in Exabyte 8500 format using compression.

ADSM Device Types

Use the 4 mm, 8 mm, QIC, an DLT device types only when the ADSM-supplied device driver is used to access the drive. If native NT device drivers are used, you must use the GENERICTAPE device type.

ADSM supports two types of libraries, manual and automated.

3.1.2.1 Manual Library

A manual library is defined for drives that you load manually. ADSM issues messages to the console when it wants to have a volume mounted into a drive. You load the volume in the drive, and ADSM starts using it. To define manual libraries to ADSM, use a LIBRARYTYPE=MANUAL statement in the library definitions.

3.1.2.2 Automated Library

An automated library is defined for drives that use automation to mount tape volumes. These libraries are defined to ADSM using a LIBRARYTYPE=SCSI statement. When defining an automated library, you must supply the ADSM device name of the autochanger mechanism. An automated library can be defined only if the ADSM-supplied device driver is used. There is no support for automated libraries if native NT device drivers are used.

The library defines the method by which tape volumes are physically mounted on drives. Associated with the library are one or more drive definitions.

3.1.2.3 Drives

Within libraries are drives, which require device class, library, and drive definitions. A drive definition is required for each drive that can mount removable media. Each drive is defined individually to ADSM. The drives are associated with a single library by the library name. The drive entry contains the ADSM device name. Figure 31 shows the definitions used to represent drives in an automated library and their relationship.

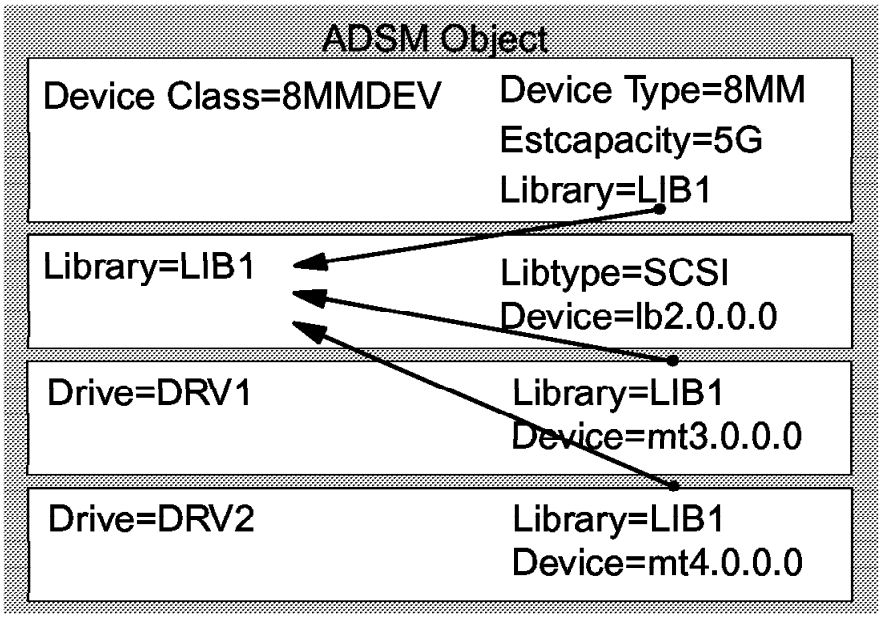


Figure 31. Library and Drive Definitions

In this example we have an 8 mm tape library. The device class, 8MMDEV, defines the type of device, its media capacity, and the library in which it is used, LIB1. The library, LIB1, defines how the library loads tape volumes. In

this example it is an automated library, libtype=SCSI, and the device name of the autoloader is lb2.0.0.0. There are two tape drives in the library, DRV1 and DRV2, each of which has its device name defined, mt3.0.0.0 and mt4.0.0.0, respectively.

3.1.3 Removable File Systems

A removable file system is a Windows NT file system that resides on a removable device. Removable file systems appear as local file systems to the NT operating system. iomega Zip and Jaz drives are examples of removable file systems supported by ADSM. The floppy disk drive (A:) is not supported because of its limited capacity.

A removable file system is defined to ADSM as a manual library through a device class with a device type of REMOVABLEFILE. ADSM treats removable file systems as if they were tape volumes. When ADSM wants to access a removable file system, it issues a mount message for the "volume" and waits for you to manually make the file system available. You have to mount (insert) the removable volume into the drive when prompted before the server can use it. ADSM maintains an inventory of all removable file system volumes in a similar manner to that for tape volumes. To identify each removable file volume, ADSM requires it to be labeled. Using the NT label command, you must assign a unique label to each volume that ADSM will use.

3.2 Removable Storage Media

ADSM represents storage media as volumes in storage pools. Volumes are grouped into storage pools, and each storage pool represents one kind of media, for example, 8 mm or 4 mm tapes (Figure 32).

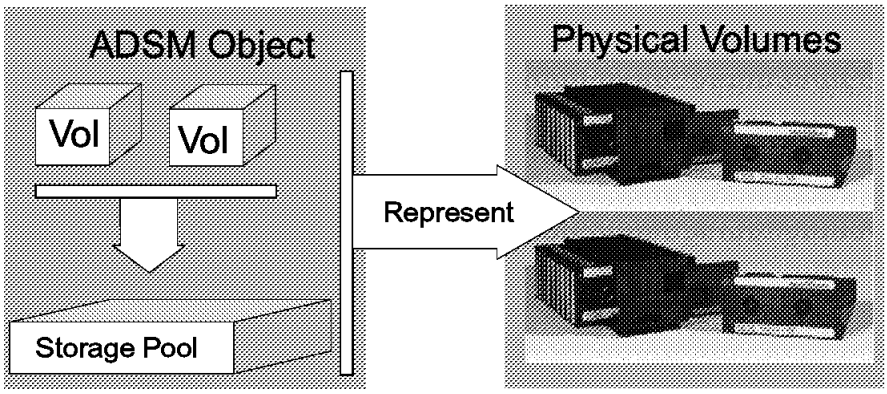


Figure 32. Media Representation

A storage pool volume is associated with one, and only one, storage pool. However, you can define multiple storage pools, containing different types of volumes. When installed, the ADSM server has three predefined storage pools: ARCHIVEPOOL, BACKUPPOOL, and SPACEMGPOOL. The storage pool volumes can be disk volumes, sequential files on disk, tape volumes, or other removable media types. When client data is sent to the server, it is associated with a storage pool through management class definitions. In this way you can differentiate where data is placed. For example, you might configure one storage pool for backups and one storage pool for archive copies of data, allowing for separation of backup and archive data.

The storage pool definitions contain information such as a pointer to the associated device class, the maximum file size allowed, and pointers to the next storage pool to use when this one is full. The storage pool volume definitions contain information such as the associated storage pool and the access mode of the volume (Figure 33).

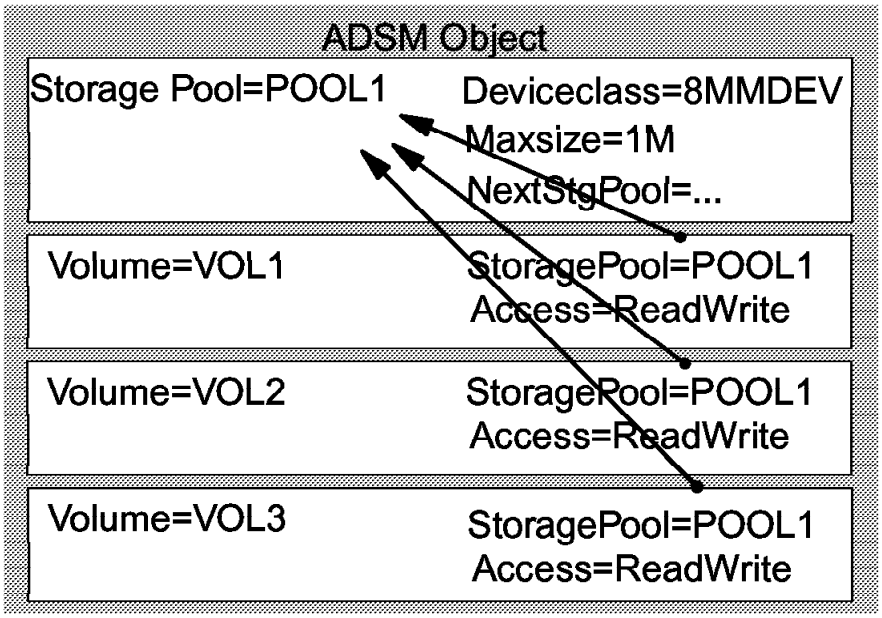


Figure 33. Storage Pool and Volume Definitions

ADSM storage pools can use either private volumes or scratch volumes. When a volume is preassigned to a storage pool, it is called a *private volume*. When a volume is not preassigned, it is called a *scratch volume*. You can choose to use either private or scratch volumes. Below we look at examples of private volume and scratch volume use.

Figure 34 on page 41 shows an example of the use of private volumes. We have two storage pools: POOL1 and POOL2. Each storage pool has a set of five volumes. POOL1 has volumes P1V1 through P1V5, and POOL2 has volumes P2V1 through P2V5. Volumes with cross-hatching are full, and volumes with a white background are empty. Assume POOL1 is used daily for backups and POOL2 is used sporadically for archives. The next time a backup runs, it will receive an out-of-space condition because all volumes in POOL1 are full. There are empty volumes in POOL2, but POOL1 cannot use them because they are private volumes assigned to POOL2.

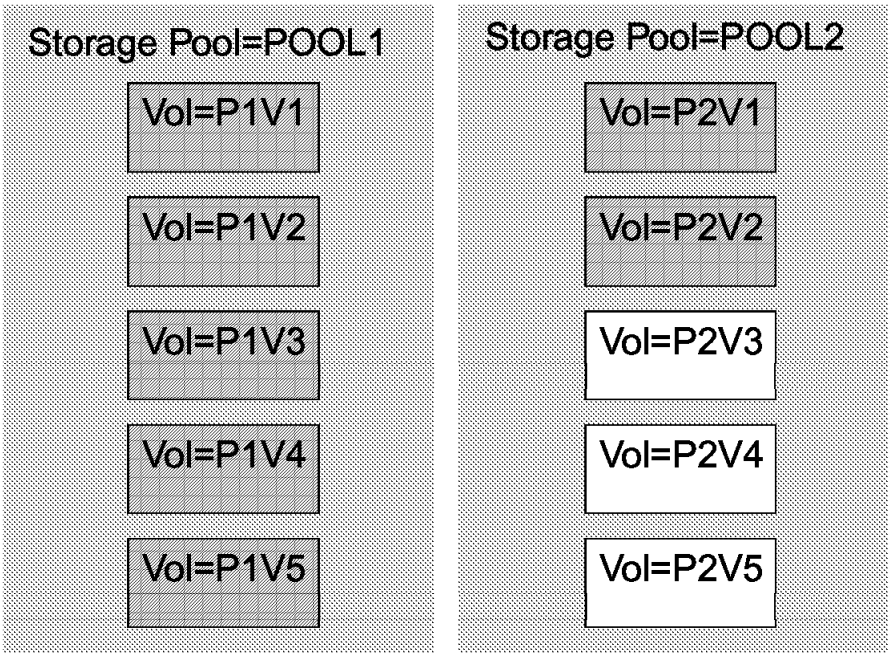


Figure 34. Using Private Storage Pool Volumes

Figure 35 on page 42 shows an example of the use of scratch volumes. We still have two storage pools, POOL1 and POOL2, but this time we also have a scratch tape pool with empty scratch volumes. POOL1 has two volumes, V1 and V5. Volume V5 has become empty because the remaining data on it has expired, so it is returned to the scratch tape pool automatically. Later, when required, a new volume V4 is assigned to POOL1 because there is new backup data to be stored. POOL2 has two full volumes, V2 and V3. When more space is needed for archive data, a new volume is dynamically added to the pool.

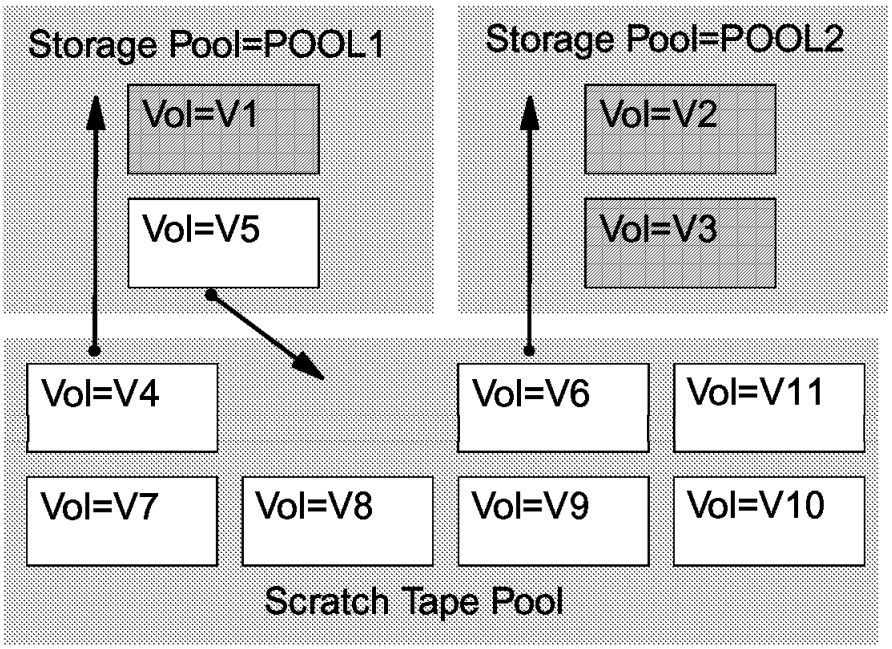


Figure 35. Using Scratch Storage Pool Volumes

Storage pools that use scratch volumes are more efficient in the use of tape than those that use private volumes because tapes are dynamically assigned to a storage pool when needed and returned to the scratch pool when empty.

3.2.1 Association of Volumes to Drives

When ADSM wants to read or write a removable media storage pool volume, it has to determine in which drive the volume can be mounted. To read an existing volume in a storage pool, the server goes through the following sequence:

1. From the storage pool and volume definitions, it determines the device class associated with the storage pool (Figure 33 on page 40).
2. From that device class, it determines the library and drives associated with that library (Figure 31 on page 38).
3. It selects a free drive if one is available, or waits until a drive becomes free.
4. It mounts the volume in the free drive.

If the server wants to write data to a storage pool, the process is similar. It determines from its database whether space is available on the existing

storage pool volumes. If space is available, the server selects a volume with available space and goes through the same sequence of steps as above, this time opening the volume for output. If space is not available, the server either:

Mounts a new private volume that has been predefined in the storage pool (Figure 34 on page 41)

or

Takes a new scratch volume and adds it to the storage pool (Figure 35 on page 42)

In either case the steps involved in mounting the volume in a drive are the same. The server uses the device class for the storage pool to determine the library and drive to use.

Chapter 4. Removable Media Configuration

In this chapter we present examples of configuring removable media devices for use by ADSM. The examples are based on our hardware configuration (Figure 36):

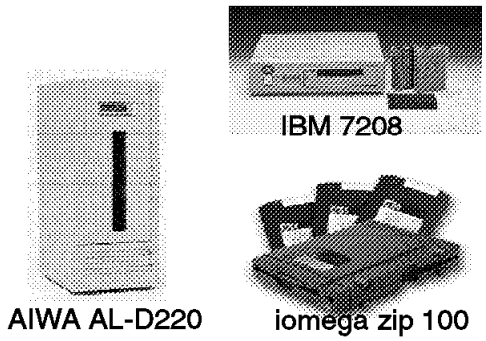


Figure 36. Removable Media Hardware Configuration

- The IBM 7208 is a stand-alone 8 mm tape drive (Exabyte 8505)
- The Aiwa AL-D220 is an automated tape library with two 4 mm tape drives and 17 slots
- The iomega zip 100 is a removable file system device with 100 MB cartridges

We also cover defining sequential files on disk. Although sequential files are not truly removable, ADSM treats them as if they are.

Examples

In all subsequent configuration examples, the appropriate ADSM commands are used at either the server console or with an administrative command line client. Many of the functions can be performed graphically with either configuration wizards or the GUI administrative client. We use commands to better illustrate the concepts and sequence of the tasks.

4.1 NT Device Drivers

Windows NT device drivers run as services. NT provides, within the Control Panel application, a Devices applet to manage device drivers and device driver startup at system boot time (Figure 37 on page 46).

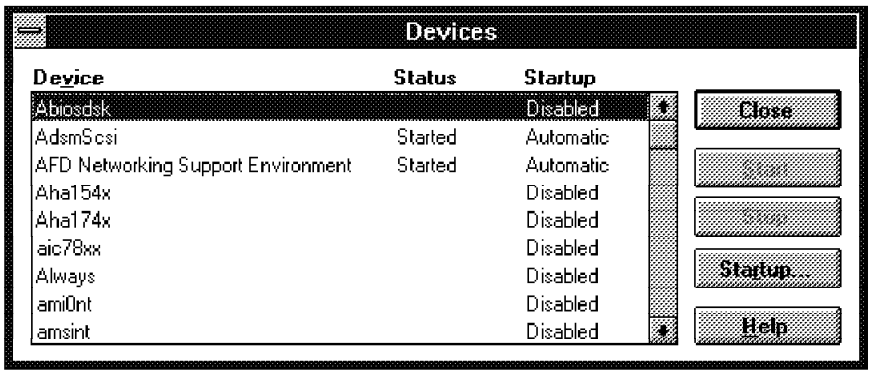


Figure 37. Devices Applet

With the Devices applet you can check the status of a device driver, start and stop devices, and change the sequence in which they start. ADSM supplies the ADSMSCSI device driver, the second device shown in Figure 37. As you can see, ADSMSCSI is started and its startup is set to "Automatic." You can alter the startup sequence for a device (see Figure 38) by clicking on the **Startup...** button in the Devices applet.

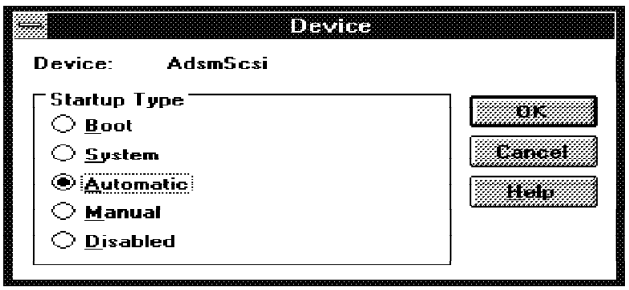


Figure 38. Device Driver Startup Settings

At boot time Windows NT starts its device drivers in the following order:

1. Boot devices
2. System devices
3. Automatic devices

To start devices listed as Manual under Startup in the Devices applet, select the device and click on **Start**. Figure 39 on page 47 shows the NT exabyte2 device driver for our 7208 drive in manual mode.

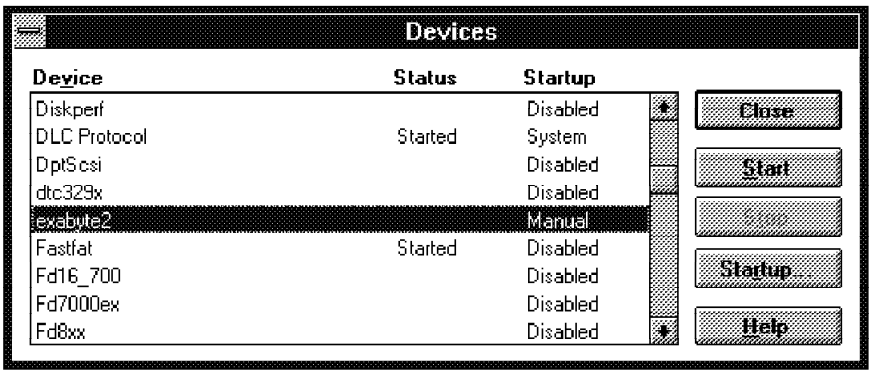


Figure 39. Exabyte2 Device Driver in Manual Mode

The driver is set to Manual as the device is being used by the ADSMSCSI device driver. Later in this section we will show how different devices can be configured to use a combination of native NT and ADSMSCSI device drivers.

4.1.1 ADSM Device Driver

The ADSM-supplied device driver, ADSMSCSI.SYS, is for SCSI-attached tape devices and automated libraries. It is the device driver of choice for all tape devices that ADSM will be use.

For tape autochanger devices, ADSM requires that you use the ADSMSCSI device driver. Removable media devices attached as local file systems require the appropriate NT device driver. Manually operated tape devices can use either the NT device driver or the ADSMSCSI device driver. The choice of device driver depends on whether the ADSM device driver supports it and whether you want to share it with other NT applications:

- If the device is supported only by the ADSMSCSI device driver, ADSMSCSI must be used for the device.
- If the device is not supported by the ADSMSCSI device driver but is supported by the NT device driver, you must configure it with a device class using a device type of GENERICTAPE.
- If the device is supported by both the ADSMSCSI device driver and NT device driver, you can choose which device driver will be used by the device:
  - If you use the ADSMSCSI device driver, the device can be used only by the ADSM server. No other NT applications can access the device.

- If you use the NT device driver, the device can be shared between the ADSM server and other NT applications.

- If you use removable media devices attached as local file systems, you must use the appropriate NT device driver and a device class with a device type of REMOVABLEFILE. The following limitations apply:
  - It must be a device with removable media.
  - It must be viewed by the operating system as a removable media drive, and not as a fixed hard disk drive.

  Devices configured with a device type of REMOVABLEFILE can be shared between ADSM and other NT applications.

ADSM cannot use all tape devices supported by NT device drivers because some of the devices may not have all of the functions that ADSM requires. For a device using an NT device driver to work correctly with ADSM, it must have at least the following capabilities:

- Write in variable mode
- Can write file marks
- Can forward and/or reverse space blocks
- Can forward and/or reverse space file marks

Care should be taken in choosing to use NT device drivers with ADSM. It is possible that a device using the ADSMSCSI driver will work properly with ADSM but display problems when using an NT device driver.

4.1.2 ADSM Device Names

ADSM uses a unique name for each device. The name can be one of the following:

1. The name that ADSMSCSI uses for the device, if the device is controlled by the ADSMSCSI driver. The name is in the following format: \\.\MTa.b.c.d or \\.\LBa.b.c.d, where:

   MT The device is a tape device

   LB The device is a controller for an automated tape library

   a SCSI ID of the device

   b Device logical unit number (LUN). Only used when multiple logical devices are attached to the same SCSI ID.

   c Bus number of the adapter. Some SCSI cards have multiple buses.

   d Port number for the SCSI adapter device driver. The port number is used by the operating system to address the SCSI card.

Figure 40 on page 49 shows an example of the ADSMSCSI addressing scheme. There are two SCSI adapters. One is port 0, and one is port 1. The port 0 adapter has two buses, 0 and 1. Two devices with SCSI IDs 5 and 6 are connected to bus 0. One device with SCSI ID 6 is connected to bus 1.

The port 1 adapter has only one bus, bus 0. One device is connected to bus 0 and uses SCSI ID 4. This device has two separately addressable logical unit numbers, LUN 0 and LUN 1.

The ADSMSCSI name for each device is shown next to the device.

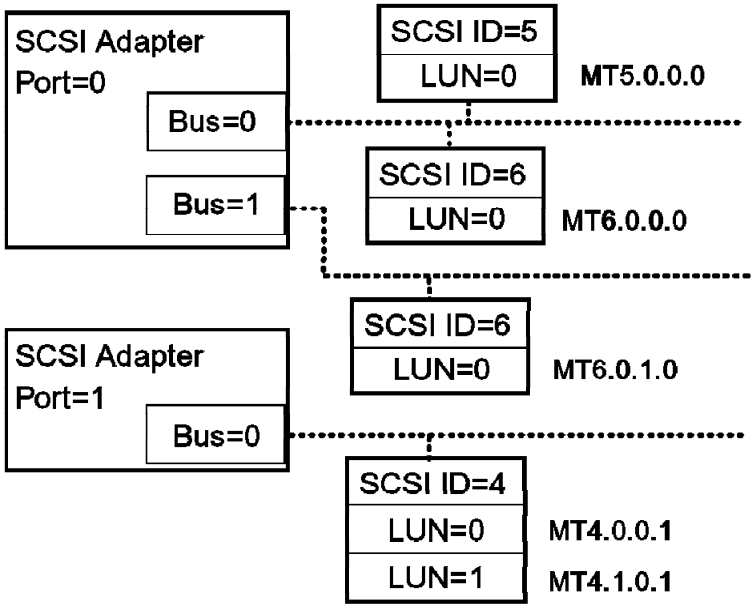


Figure 40. ADSMSCSI Device Naming Scheme

2. The name NT uses for the device, when a native NT driver is used. NT device drivers use device names such as \\.\Tape0. These names are assigned sequentially at system startup time and can change on the basis of the hardware active in the system at boot time and the startup sequence of devices. If you are using NT drivers, we recommend that you use alias names. These names do not change between system boots, provided you do not change how the device is attached to the system.
3. An alias name. The format of alias names is identical to the ADSMSCSI device names. Examples are mt3.0.0.1 and lb2.0.0.1. To distinguish between real and alias names ADSM checks for a backslash (\) at the

beginning of the name. Names beginning with a backslash indicate real device names, and names without a backslash indicate alias names.

4. A drive letter for devices attached as local removable file systems

4.1.3 Determining ADSM Device Names

ADSM supplies a diagnostic utility to determine the device name and appropriate device type. This utility gathers information from the NT registry for all SCSI-attached devices and presents it to you. From the output you can determine the device names and device types to use:

1. Open the ADSM program window
2. Select the ADSM **Server Utilities** icon
3. Select the **Diagnostics** icon
4. The utility displays two tabbed pages. The Device Detector tab contains the device information we require (Figure 41).

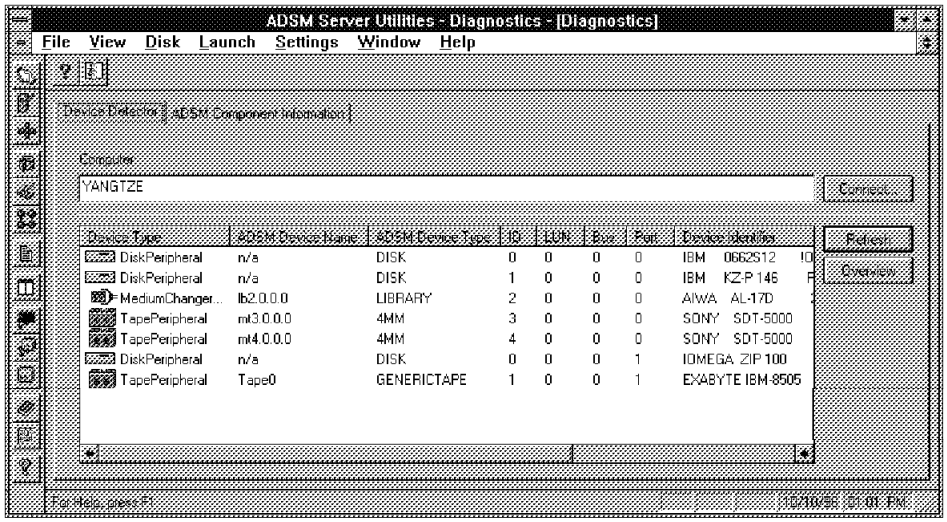


Figure 41. Diagnostics Utility: Device Detector Information

From this display we can determine which tape devices are using the ADSMSCSI device driver and the ADSM device names assigned to them. ADSMSCSI controlled tape devices will have names starting with **lb** for library autochangers and **mt** for tape drives, either within a library or manual drives. The device name is used to define the device to ADSM. Also displayed is an ADSM device type for each device. The device type is used in configuring an appropriate device class.

If the tape device is using a native NT device driver, an NT device name will be displayed in the ADSM Device Name column. Native NT device names for tapes take the form: \\.\Tape0. However, the ADSM diagnostics display only an abbreviated name, Tape0. This NT device name can be used to define to ADSM devices that use native NT device drivers. However, the NT names are assigned sequentially every time the system is restarted, according to the device startup sequence. If a new device is added, and it starts earlier than an existing device, it will be assigned its name first, and subsequent devices will have their names changed. This could cause a problem with ADSM; a device assigned using a specific name will have to be redefined to ADSM if its NT name changes.

To avoid this problem, an ADSM alias can be used for devices that native NT device drivers. An alias takes the same form as an ADSMSCSI tape device name: MTa.b.c.d. The alias can be determined by looking at the Device Detector display (Figure 41 on page 50) and using the ID, LUN, Bus, and Port columns. In this display the 7208 drive is using the exabyte2 NT device driver. The device detector has determined that its NT device name is Tape0. In this example its alias would be MT1.0.0.1.

We strongly recommend that you use an alias rather than the native NT device name. Finally, if the tape device is using an NT device driver, you must always use an ADSM device type of GENERICTAPE in defining the ADSM device class.

An alternative method of determining tape device names is to examine the SCSI device entries in the NT registry. Use the registry editor, REGEDT32.EXE, which displays four windows, one for each of the registry hives. The SCSI configuration entries are in the HKEY_LOCAL_MACHINE structure (Figure 42 on page 52).

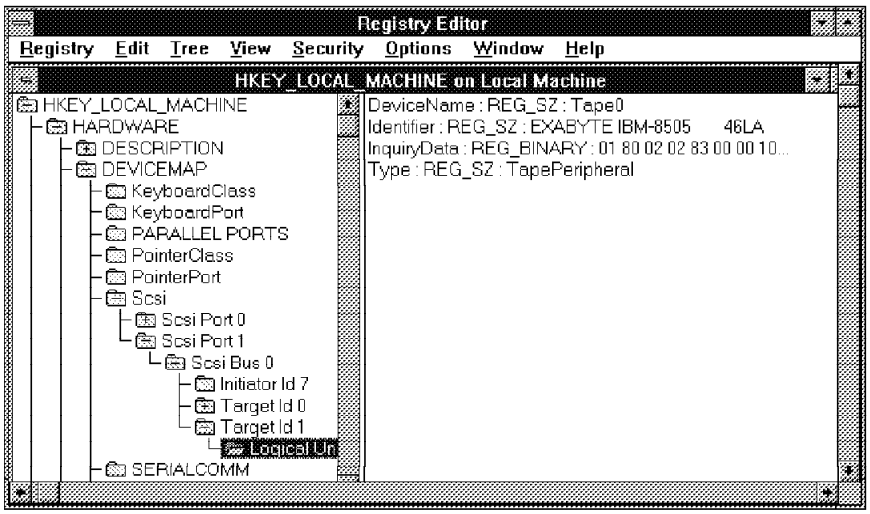


Figure 42. SCSI Device Registry Entries

Navigate down HKEY_LOCAL_MACHINE, opening the HARDWARE, DEVICEMAP, and Scsi directories. Within the Scsi directory are entries for each port and the devices installed on them. For each device there is a Logical Unit entry, which has a number of keys. The keys are displayed in the right-hand panel. The DeviceName key shows the current device name for that device. If the tape device is using an NT device driver, the name will be an NT device name such as Tape0. If the device is using the ADSMSCSI device driver, the device name will be in the ADSM format, MTa.b.c.d. For devices controlled by ADSMSCSI, an ADSMDriver key will show the date of the device driver, and a DEVCLASS key will show the ADSM device class that should be used. The same information is displayed when you use the ADSM Device Detector.

4.1.4 Configuring Multiple Device Drivers

If you have multiple tape devices connected to the ADSM server, you can choose to use NT drivers for some devices and the ADSMSCSI driver for others. In this way, devices can be shared with other applications that cannot use ADSMSCSI-controlled devices. The startup sequence of the device drivers determines the devices that are controlled by the drivers:

- When you start an NT driver, it claims, or reserves, for its use all of the devices that it supports.
- When you start the ADSMSCSI device driver, it claims all of the supported devices that it finds.

This can lead to conflicts between device drivers attempting to use a device that has already been reserved by another driver. In our configuration the

7208 drive is supported by the ADSMSCSI driver and the native exabyte2 NT driver. If both drivers are configured to start automatically, the ADSMSCSI driver reserves the drive first, causing an error when the exabyte2 driver also attempts to reserve it.

You can exclude devices that you want to use with NT device drivers from being reserved by the ADSMSCSI device driver at system startup. Open ADSM Server Utilities - Services and select the Device Driver tab. Then select the **Device Driver Options** button, and a pop-up window will appear (Figure 43).

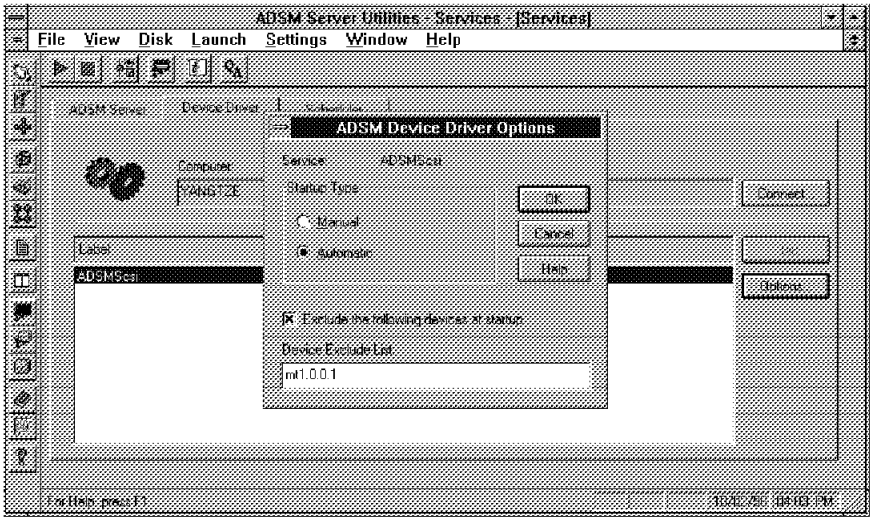


Figure 43. ADSMSCSI Configuration: Excluding Devices from Startup

Check the “Exclude the following devices at Startup” box and enter the list of devices to exclude, using their ADSM name or alias. In the example we excluded device mt1.0.0.1, the ADSM alias for our 7208 tape drive, which uses the native NT device driver, exabyte2, and is known to NT as Tape0.

Reboot the system for the changes to take effect.

4.2 Configuring Removable Media Devices

In this section we show how to configure devices using the administrative command line client. It is necessary to use this interface when issuing commands that create or update a device class, library, or drive. We assume that all devices have been SCSI attached and connected to the ADSM server system.

Before we can configure the devices we must determine their addresses, either the ADSMSCSI address or the alias if the device is using a native NT device driver. We will use the ADSM Server Utilities-Diagnostics, as discussed in 4.1.3, “Determining ADSM Device Names” on page 50. Figure 44 on page 54 shows the Device Detector output for our sample configuration.

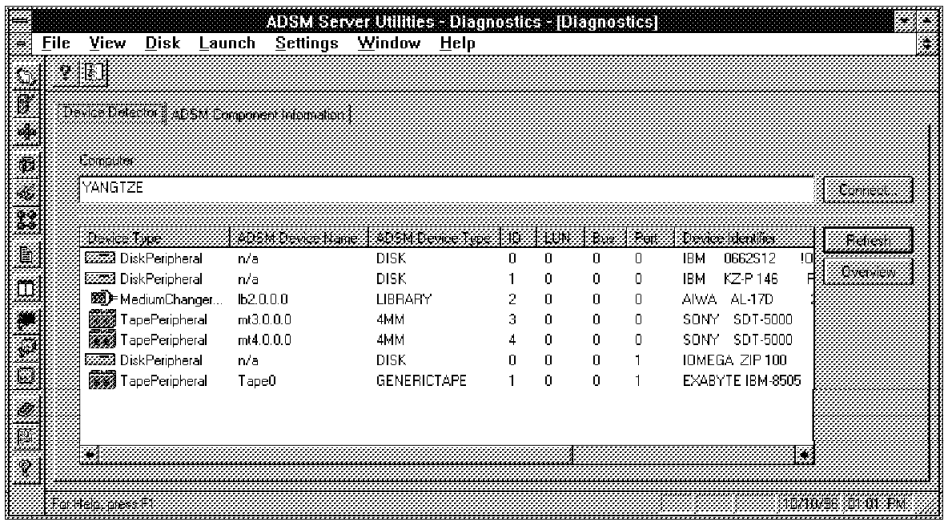


Figure 44. Sample Configuration: Device Detector Information

4.2.1 Manual Tape Library Configuration

In the example we configure the 7208 tape drive to use the NT device driver so that the drive is available to ADSM and other NT applications. Before starting, ensure that the device driver for this device is properly configured (refer to 4.1.4, “Configuring Multiple Device Drivers” on page 52).

From the Device Detector information (Figure 44) we can determine that the Windows NT device name is Tape0 and its ADSM alias is mt1.0.0.1. To define the device, using the ADSM alias, follow these steps:

1. Define a manual library with a name of 8MMLIB:

```
adsm> define library 8MMLIB libtype=manual
```

2. Define the drive that will belong to the 8MMLIB library. In our case we have only one drive, DRIVE1. The device name alias is mt1.0.0.1:

```
adsm> define drive   8MMLIB  DRIVE1  device=mt1.0.0.1
```

3. Define a device class 8MM for library 8MMLIB. The device class defines the device characteristics. In this example we define the device class with a device type of GENERICTAPE as the device is using the NT device driver:

```
adsm> define devclass  8MM  devt=generictape  library=8MMLIB  mountlimit=1
```

   We set the mount limit to 1 as we only have one drive defined in the library.

4. Verify the operations by issuing the following query commands:

```
adsm> query library

Library Name     Library        Device               External
                 Type                                Manager
------------     ----------     ----------------     --------------
8MMLIB           MANUAL

adsm> query drive

Library Name     Drive Name       Device Type     Device               Element
------------     ------------     -----------     ----------------     -------
8MMLIB           DRIVE1           GENERICTAPE     MT1.0.0.1

adsm> query devclass

Device       Device        Storage    Device       Format     Est/Max    Mount
Class        Access           Pool    Type                  Capacity   Limit
Name         Strategy        Count                             (MB)
---------    ----------    -------    ---------    ------    --------    -----
8MM          Sequential          0    GENERIC      DRIVE          0.0        1
                                       TAPE
```

5. If you want to get more detailed information about device class 8MM, issue the query devclass command again with the Format=Detailed option:

```
adsm> query devclass 8MM Format=Detailed

              Device Class Name: 8MM
        Device Access Strategy: Sequential
            Storage Pool Count: 0
                   Device Type: GENERIC TAPE
                        Format: DRIVE
         Est/Max Capacity (MB): 0.0
                   Mount Limit: 1
              Mount Wait (min): 60
         Mount Retention (min): 60
                  Label Prefix: ADSM
                  Drive Letter:
                       Library: 8MMLIB
                     Directory:
Last Update by (administrator): ADMIN
         Last Update Date/Time: 09/30/1996 19:17:30
```

4.2.2 Automatic Tape Library Configuration

The Aiwa automated tape library contains two 4 mm drives and is supported only for use with the ADSMSCSI device driver. Therefore it can only be used by ADSM.

We can determine the ADSM device names from Device Detector information (Figure 44 on page 54):

lb2.0.0.0 Name of the autochanger
mt3.0.0.0 Name of first drive
mt4.0.0.0 Name of second drive

Next we have to determine the association between the ADSM address and the element address. We need this association when defining the drives.

The element address is the internal hardware address for the individual addressable components of a library: cartridge slots, drives, and autochanger. These element addresses are 2-byte hexadecimal addresses, assigned in the device hardware by the manufacturer. Each make of library uses different element addresses for its various components. The *ADSM for Windows NT Administrator's Guide* contains worksheets for all supported libraries. These worksheets contain the element addresses that must be used when defining that make of library. Figure 45 on page 57 shows an example of the worksheet for the Aiwa AL-D220 with the SCSI IDs and ADSM device names filled in at the top. The lower part or the worksheet lists the element addresses for the various library components.

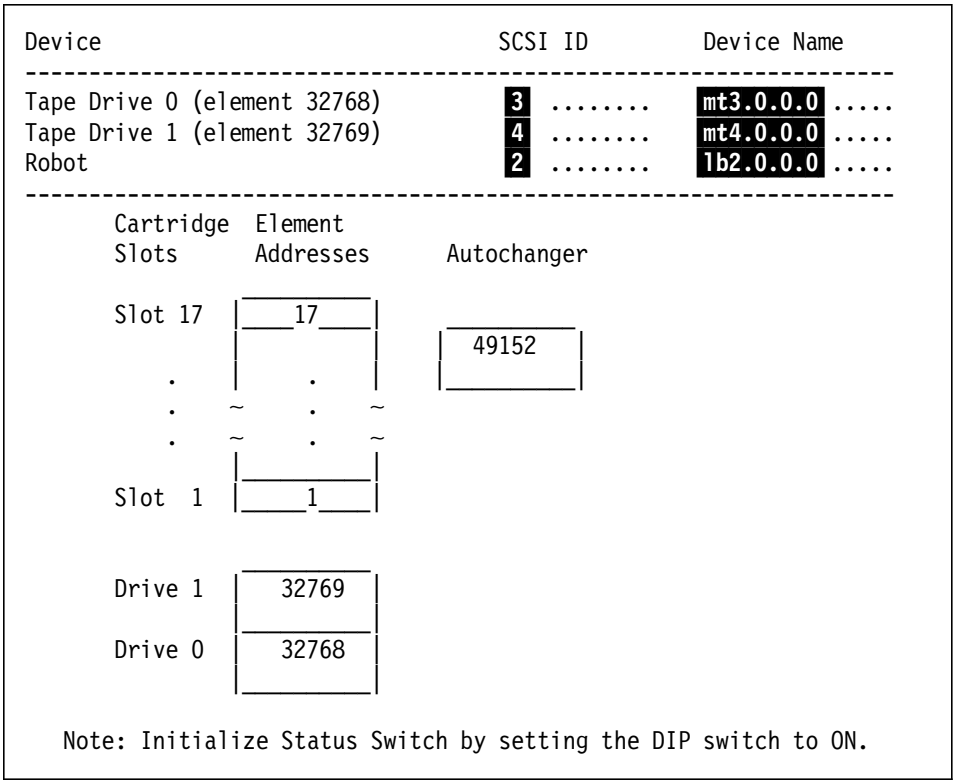


Figure 45. Configuration Worksheet for Aiwa AL-D220

With this information from the worksheet, you are now ready to define the devices. Follow these steps:

1. Define a library to ADSM called 4MMLIB. We define a SCSI library, and the ADSM device name is lb2.0.0.0. We abbreviate the name to lb2 as trailing zeroes can be omitted:

```
adsm> define library 4MMLIB libtype=SCSI device=lb2
```

2. Define two drives, DRIVE0 and DRIVE1, in the 4MMLIB library, using the abbreviated ADSM device name and element address from the completed worksheet shown in Figure 45:

```
adsm> define drive 4MMLIB DRIVE0 device=mt3 element=32768
adsm> define drive 4MMLIB DRIVE1 device=mt4 element=32769
```

3. Define the ADSM device class for the drives. We create a device class for our 4MM drives called 4MM, which points to library 4MMLIB. We use the highest recording capacity the hardware supports by specifying

format=drive. We want a maximum of two tapes to be mounted concurrently, as we have two drives, so we specify mountlimit=2. We used the following command:

```
adsm> define devclass 4MM devtype=4mm format=drive libr=4MMLIB mountlimit=2
```

4. Verify the operations by issuing the appropriate query commands:

```
adsm> query library 4MMLIB

Library Name     Library        Device               External
                 Type                                Manager
------------     ----------     ----------------     --------------
4MMLIB           SCSI           LB2

adsm> query drive 4MMLIB

Library Name     Drive Name       Device Type     Device               Element
------------     ------------     -----------     ----------------     -------
4MMLIB           DRIVE0           4MM             MT3                   32,768
4MMLIB           DRIVE1           4MM             MT4                   32,769

adsm> query devclass 4MM

Device       Device        Storage    Device       Format     Est/Max    Mount
Class        Access           Pool    Type                  Capacity    Limit
Name         Strategy        Count                             (MB)
---------    ----------    -------    ---------    ------    --------    -----
4MM          Sequential          0    4MM          DRIVE          0.0        2
```

4.2.3 Removable File System Configuration

In this example we configure the iomega zip 100 drive as a REMOVABLEFILE device type to ADSM and call the device class ZIP. The iomega zip 100 device is defined to our NT system as the F: drive. It is not accessed by its SCSI address but by its drive letter. To define the removable files, follow these steps:

1. Define a library called ZIPLIB. This is a manual library as we have no autochanger mechanism.

```
adsm> define library ZIPLIB libtype=manual
```

2. Define a drive, ZIP_DRIVE, within the ZIPLIB library. The drive will use the NT F: drive.

```
adsm> define drive ZIPLIB ZIP_DRIVE device=F:
```

3. Define the ZIP device class with the REMOVABLEFILE device type. Associate it with library ZIPLIB.

```
adsm> define devclass ZIP devt=REMOVABLEFILE libr=ZIPLIB
```

4. Confirm these definitions by issuing these query commands:

```
adsm> query library ZIPLIB

Library Name     Library        Device               External
                 Type                                Manager
------------     ----------     ----------------     --------------
ZIPLIB           MANUAL

adsm> query drive ZIPLIB

Library Name     Drive Name       Device Type     Device               Element
------------     ------------     -----------     ----------------     -------
ZIPLIB           ZIP_DRIVE        REMOVABLE-      F:
                                   FILE

adsm> query devclass ZIP

Device       Device        Storage    Device       Format     Est/Max    Mount
Class        Access           Pool    Type                   Capacity   Limit
Name         Strategy        Count                              (MB)
---------    ----------    -------    ---------    ------    --------    -----
ZIP          Sequential          0    REMOVABLE                   4.0        1
                                       FILE
```

4.2.4 Sequential File Configuration

In this example we define a device class with a FILE device type and call it FILE. This device class will create sequential files and put them in the D:\ADSMFILE directory on a local server disk. Note that this drive could also be a remote drive accessed through either TCP/IP NFS or a network drive. We want the files on this drive to be no larger than 500 KB so we specify maxcap=500k. The command to create this device class is:

```
adsm> define devclass FILE devtype=FILE maxcap=500k dir=D:\ADSMFILE
```

After defining the device class, you can query the server for the definition again, using the QUERY DEVCLASS command and the format=detailed parameter to see the capacity and directory information:

```
adsm> query devclass FILE format=detailed

             Device Class Name: FILE
        Device Access Strategy: Sequential
            Storage Pool Count: 0
                   Device Type: FILE
                        Format:
         Est/Max Capacity (MB): 0.5
                   Mount Limit: 1
              Mount Wait (min):
         Mount Retention (min):
                  Label Prefix:
                  Drive Letter:
                       Library:
                     Directory: D:\ADSMFILE
Last Update by (administrator): ADMIN
         Last Update Date/Time: 10/01/1996 13:21:52
```

4.2.5 The Device Configuration File

ADSM maintains device configuration within its database. For availability and recovery in the case of a database failure, a copy of the device configuration can be created in one or more device configuration files. The device configuration files are defined with the DEVCONFIG option in the ADSM server options file: DSMSERV.OPT. You can assign any name to the device configuration files, and you can display the files by using the QUERY OPTION command:

```
adsm> query option

Server Option     Option Setting       Server Option     Option Setting
----------------- -------------------- ----------------- --------------------
CommTimeOut       60                   IdleTimeOut       15
BufPoolSize       512                  LogPoolSize       128
DateFormat        1 (mm/dd/yyyy)       TimeFormat        1 (hh:mm:ss)
NumberFormat      1 (1,000.00)         MessageFormat     1
Language          AMENG                MaxSessions       25
ExpInterval       1                    ExpQuiet          No
MirrorRead DB     Normal               MirrorRead LOG    Normal
MirrorWrite DB    Sequential           MirrorWrite LOG   Parallel
VolumeHistory     volhist.out          Devconfig         devcnfg.out
TxnGroupMax       16                   MoveBatchSize     32
MoveSizeThresh    1                    StatusMsgCnt      1
TcpPort           1500                 TCPWindowsize     8192
TCPNoDelay        No                   IPXSocket         8522
NetBiosName       ADSMNT               NetbiosBufferSize 16384
NetbiosSessions   16                   IpxBufferSize     32768
NamedPipeName     \\.\PIPE\ADSMPIPE    LanAdapter        0
CommMethod        TCPIP                CommMethod        NETBIOS
CommMethod        NAMEDPIPE            Message Interval  1
MaxMemory         536870912
```

In the above example, we use the default name, DEVCNFG.OUT, which is a text file stored by default in the server directory together with the options file and other server files. DEVCNFG.OUT contains the commands necessary to re-create the server device configuration:

```
/* IBM AdStar Distributed Storage Manager Device Configuration */
DEFINE DEVCLASS 4MM DEVTYPE=4MM FORMAT=DRIVE MOUNTLIMIT=2 MOUNTWAIT=60...
                                    ...MOUNTRETENTION=60 PREFIX=ADSM LIBRARY=4MMLIB
DEFINE DEVCLASS FILE DEVTYPE=FILE MAXCAPACITY=4096K MOUNTLIMIT=1 DIRECTORY=D:\ADSMFILE
DEFINE DEVCLASS 8MM DEVTYPE=8MM FORMAT=DRIVE MOUNTLIMIT=1 MOUNTWAIT=60...
                                    ...MOUNTRETENTION=60 PREFIX=ADSM LIBRARY=8MMLIB
DEFINE DEVCLASS ZIP DEVTYPE=REMOVABLEFILE MAXCAPACITY=4096K MOUNTLIMIT=1...
                                   ...MOUNTWAIT=60 MOUNTRETENTION=60 LIBRARY=ZIPLIB
DEFINE LIBRARY 4MMLIB LIBTYPE=SCSI DEVICE=LB2
DEFINE LIBRARY 8MMLIB LIBTYPE=MANUAL
DEFINE LIBRARY ZIPLIB LIBTYPE=MANUAL
DEFINE DRIVE 4MMLIB DRIVE0 DEVICE=MT3 ELEMENT=32768
DEFINE DRIVE 4MMLIB DRIVE1 DEVICE=MT4 ELEMENT=32769
DEFINE DRIVE 8MMLIB DRIVE1 DEVICE=MT1.0.0.1
DEFINE DRIVE ZIPLIB ZIP_DRIVE DEVICE=F:
```

The device configuration file is automatically updated as and when the device configuration is changed. Any define or update commands that alter the device configuration are automatically reflected in the file. These automatic updates are made to the file names defined in DSMSERV.OPT. In addition to this automatic process, you can back up the device configuration, using the BACKUP DEVCONFIG administrator command:

```
adsm> backup devconfig file=devbak.txt
```

In the above example, the device configuration is backed up to the specified file. If you do not use the file= parameter, the configuration will be backed up to the file names specified in DSMSERV.OPT.

The device configuration file is vital for recovery of the server in case of database loss. It is used during a database recovery to configure the devices so that a database backup volume can be read.

4.3 Configuring Storage Pools

A storage pool is a collection of volumes that belong to the same device class. ADSM uses storage pools to store data from its clients. Storage pools can be “chained” together to create a hierarchy of storage pools using different types of devices. In this section we define a set of storage pools that use the removable media devices configured in 4.2, “Configuring Removable Media Devices” on page 53.

The ADSM server defines three storage pools on disk devices during installation. The installation program defines one disk volume in the BACKUPPOOL storage pool:

```
adsm> query volume

Volume Name               Storage      Device      Estimated  %Util   Volume
                          Pool Name    Class Name   Capacity          Status
                                                       (MB)
------------------------  -----------  ----------  ---------  -----  --------
C:\WIN32APP\IBM\ADSM\SE-  BACKUPPOOL   DISK              4.0   73.0  On-Line
 RVER\DATA1.DSM
```

You can define additional disk storage pool volumes as discussed in 2.2.5, “Formatting Server Disk Volumes” on page 23. Each volume is associated with a storage pool. When all volumes in a storage pool are full, ADSM cannot store any more data into that storage pool. To view the overall structure of the storage pools, use the QUERY STGPOOL command:

```
adsm> query stgpool

Storage      Device       Estimated  %Util  %Migr  High   Low  Next
Pool Name    Class Name    Capacity                Mig%  Mig%  Storage
                               (MB)                          Pool
-----------  ----------  ----------  -----  -----  ----  ----  ----------
ARCHIVEPOOL  DISK               4.0    0.0    0.0    90    70
BACKUPPOOL   DISK               4.0   73.0   73.0    90    70
SPACEMGPOOL  DISK               0.0    0.0    0.0    90    70
```

Notice the Next Storage Pool column. ADSM allows you to migrate data to another storage pool when the current storage pool fills above a certain limit. We will exploit this capability to move from disk to the Aiwa library.

In the rest of this section we set up and use the configured devices and their device classes. Figure 46 on page 63 shows our storage pool structure. During this project we did not use the ADSM SPACEMGPOOL, so we did not configure that pool any further.

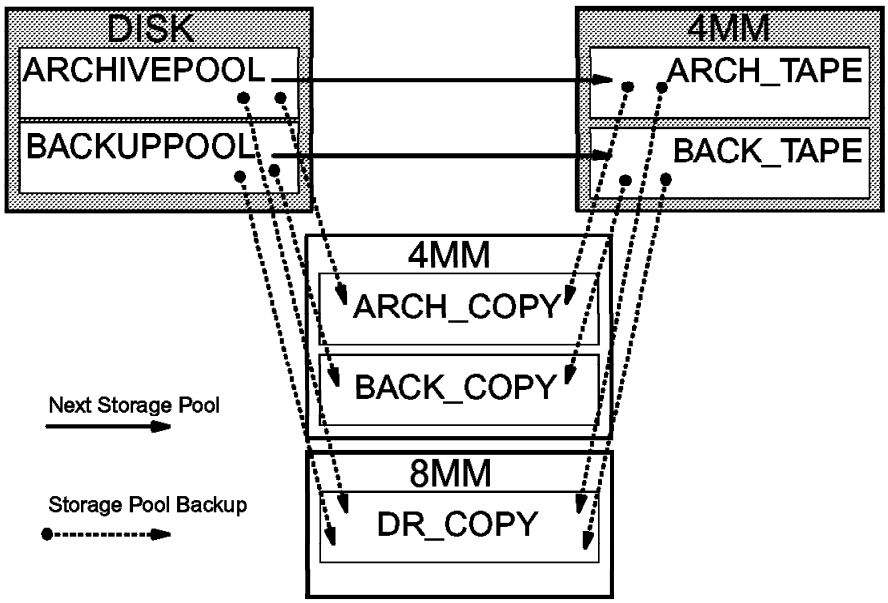


Figure 46. Storage Pool Configuration

As you can see we have two storage pools, ARCHIVEPOOL and BACKUPPOOL, on disk using the DISK device class. We define two tape storage pools using the 4MM device class and update the next storage pool pointer in ARCHIVEPOOL and BACKUPPOOL to migrate data onto the new storage pools.

Note

Figure 46 shows the configuration of the primary and copy storage pools to illustrate the point that multiple storage pools can be configured using different device classes. Configuration and use of copy storage pools are covered in Chapter 6, “Storage Pool Availability” on page 97.

To configure the primary storage pools, follow these steps:

1. Define two primary storage pools, BACK_TAPE for backup data and ARCH_TAPE for archive data, to use the 4MM device class:

```
adsm> define stgpool BACK_TAPE 4MM maxscr=60 reclaim=60
adsm> define stgpool ARCH_TAPE 4MM maxscr=60 reclaim=60
```

   The two storage pools are mapped to the same device class and use scratch volumes that have been inserted into the library. The MAXSCR

parameter tells ADSM to use scratch volumes for the storage pool. The reclaim parameter controls reclamation of partially empty tapes. With reclaim=60, when the tape has more than 60% of free space on it, a reclamation process will start and move the data to another volume.

2. Update the disk storage pools to point to the tape storage pools. Data will be migrated from the disk pool to the tape pool according to the occupation of the disk storage pools and the migration thresholds. In this example, data will be migrated from BACKUPPOOL to BACK_TAPE when BACKUPPOOL is more than 80% full:

```
adsm> update stgpool BACKUPPOOL nextstgpool=BACK_TAPE hi=80 lo=60
adsm> update stgpool ARCHIVEPOOL nextstgpool=ARCH_TAPE hi=80 lo=60
```

3. Verify the previous commands by issuing a QUERY STGPOOL command:

```
adsm> query stgpool

Storage      Device       Estimated  %Util  %Migr  High   Low  Next
Pool Name    Class Name    Capacity                Mig%  Mig%  Storage
                               (MB)                        Pool
-----------  ----------  ----------  -----  -----  ----  ----  -----------
ARCHIVEPOOL  DISK               4.0    0.0    0.0    80    60  ARCH_TAPE
ARCH_TAPE    4MM                0.0    0.0    0.0    90    70
BACKUPPOOL   DISK               4.0   73.0   73.0    80    60  BACK_TAPE
BACK_TAPE    4MM                0.0    0.0    0.0    90    70
SPACEMGPOOL  DISK               0.0    0.0    0.0    90    70
```

The output of the command shows that ARCHIVEPOOL data will migrate to the ARCH_TAPE storage pool and BACKUPPOOL data will migrate to the BACK_TAPE pool. Both of these pools will use the Aiwa library because we used the 4MM device class.

4.4 Removable Media Preparation

ADSM requires that all volumes under its control be labeled. ADSM manages its inventory of volumes by name and uses the volume label as the volume name. The volume label is a special file written at the beginning of the tape that contains this name. The name can be up to six characters long. ADSM supplies a utility to write a label to tape media volumes.

Tape volumes for use in an automated library must be checked into the library so that ADSM can synchronize its library inventory with that of the physical library.

4.4.1 Tape Labeling

ADSM supplies the DSMLABEL utility, which is executed from a command prompt, and the labeling wizard, which is started from the ADSM Server Utilities. In our examples we use the DSMLABEL command, which is located in the ADSM utilities directory, C:\WIN32APP\IBM\ADSM\UTILS.

The DSMLABEL command requires at least the ADSM address of the device that will be used to label the volumes. Additional options enable you to specify a library that contains the drive or drives, whether you will use barcode support, whether you want to overwrite the existing label, and so on. Below we label and prepare volumes for use with all of the devices that we have defined.

4.4.1.1 Labeling in a Manual Library

To label an 8 mm volume on the 7208 tape drive, enter the following at a command prompt:

```
dsmlabel -drive=mt1.0.0.1 -overwrite
```

Specify the overwrite option to label the tape regardless of its content. If you omit the overwrite option, the DSMLABEL utility will not label the volume. After you issue the DSMLABEL command, the ANR9722I message prompts you to insert a new volume into the drive, supply a volume label name to the DSMLABEL utility, and press Enter to start the labeling operation. When the labeling operation terminates, message ANR9720I informs you that the operation was successful, and message ANR9722I prompts you for another volume to label. If you want to stop, press Enter without supplying a volume name.

```
ADSTAR Distributed Storage Manager for Windows NT
Volume Label Utility Program
Version 2, Release 1, Level 0.0/1.0
Licensed Materials - Property of IBM

5639-A09 (C) Copyright IBM Corporation 1994, 1996.  All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR9722I Insert a new volume in drive 'MT1.0.0.1', then enter the name (1-6
characters) to be used for its label; or just press ENTER to quit this program:
COPY01
ANR9743I Attempting to label volume 'COPY01' using drive 'MT1.0.0.1'...
ANR9720I Volume 'COPY01' was labeled successfully using drive 'MT1.0.0.1'.

ANR9722I Insert a new volume in drive 'MT1.0.0.1', then enter the name (1-6
characters) to be used for its label; or just press ENTER to quit this program:
```

Volume COPY01 can now be used as a copy storage pool volume.

4.4.1.2 Labeling in an Automated Library

Labeling volumes for use in an automated library differs from labeling volumes in a manual tape drive in that you can insert a group of volumes and use the -search parameter to have the DSMLABEL utility search the library for the volumes that are present and label them. You must supply the library name, drive name, and element addresses. To overwrite an existing label, use the -overwrite parameter.

```
dsmlabel -drive=mt3,32768 -library=lb2 -search -overwrite
```

Use the -search and -overwrite parameters together *only* if the library does not currently contain any ADSM volumes. If the library contains volumes that ADSM is using, remove the -overwrite parameter. You can still use the -search option; only new tapes will be labeled.

You are prompted for volume labels, but the library automatically performs the mounts.

```
ADSTAR Distributed Storage Manager for Windows NT
Volume Label Utility Program
Version 2, Release 1, Level 0.0/1.0
Licensed Materials - Property of IBM
5639-A09 (C) Copyright IBM Corporation 1994, 1996.  All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR9739I Initializing library device 'LB2'.

ANR9735I Enter the name (1-6 characters) for the next volume to label in
library 'LB2';  or just press ENTER to quit this program:
LIBR01
ANR9743I Attempting to label volume 'LIBR01' using drive 'MT3'...
ANR9720I Volume 'LIBR01' was labeled successfully using drive 'MT3'.

ANR9735I Enter the name (1-6 characters) for the next volume to label in
library 'LB2';  or just press ENTER to quit this program:
LIBR02
ANR9743I Attempting to label volume 'LIBR02' using drive 'MT3'...
ANR9720I Volume 'LIBR02' was labeled successfully using drive 'MT3'.

ANR9735I Enter the name (1-6 characters) for the next volume to label in
library 'LB2';  or just press ENTER to quit this program:
```

We suggest that you write external tape labels that match the internal tape labels on the volumes. One way of ensuring that the external tape label is the same as the internal ADSM tape label is to use the following parameters:

```
dsmlabel -drive=mt3,32768 -library=lb2 -overwrite
```

In this example DSMLABEL will not search the library. It prompts you to insert a new volume into the library and supply the label name. Then the volume is labeled and you are prompted to remove it from the library.

Another way of ensuring that external tape labels match the internal tape labels is to use the ADSM barcode support and barcode labels as discussed in 4.4.3, “Autochanger Barcode Support” on page 69.

4.4.2 Maintaining Library Inventories

After the volumes have been labeled, you must inform the ADSM server of the volumes that are available in a library using the CHECKIN LIBVOLUME command. You can use the AUDIT LIBRARY command to verify the library inventory and the CHECKOUT LIBVOLUME command to eject a volume from a library. You can issue these commands from the ADSM console, if available, or from an ADSM administrative command line client.

4.4.2.1 Checking Volumes into the Library

You must inform the server of the volumes you labeled previously and inserted into the Aiwa AL-D220 library. This library is known to ADSM as 4MMLIB. You want to define the volumes as scratch volumes so they can be dynamically added to the storage pools that need a new volume, as discussed in 3.2, “Removable Storage Media” on page 39. Use the status=scratch parameter. The search=yes parameter tells ADSM to search the library for all volumes. As ADSM will start a background process, we suggest that you start an ADSM administrative command line client using the -consolemode option. In this way ADSM will display all messages to this client. Issue the following command:

```
adsm> checkin libvolume  4MMLIB  status=scratch  search=yes
```

ADSM starts a background process. To monitor the process, use the QUERY PROCESS command and check the ADSM console for error messages. ADSM checks the label of every volume by mounting it on a drive and reading the label. When the process ends, you can verify which tapes have been checked in by issuing the QUERY LIBVOLUME command, which displays the ADSM inventory of the library:

```
adsm> query libvolume 4MMLIB

Library Name     Volume Name     Status                Last Use
------------     -----------     ------------------    ---------
4MMLIB           LIBR01          Scratch
4MMLIB           LIBR02          Scratch
4MMLIB           LIBR03          Scratch
4MMLIB           LIBR04          Scratch
```

It is also possible to inform ADSM of individual volumes. If you want to insert a new volume, for example, LIBR05, into the library and use it as a scratch volume, issue the following command:

```
adsm> checkin libvolume 4MMLIB LIBR05 STAT=SCRATCH
```

In this case, as the search=yes option is not used, ADSM prompts you to insert the volume into an empty slot. It then checks the label of the volume, and, if it matches the label specified with the command, it accepts the volume:

```
ANR8422I CHECKIN LIBVOLUME: Operation for library 4MMLIB started as process
43.
ANR8306I 002: Insert 4MM volume LIBR05 R/W into the slot with element number 9
of library 4MMLIB within 60 minutes; issue 'REPLY' along with the request ID
when ready.
reply 2
ANR8335I 002: Verifying label of 4MM volume LIBR05 in drive DRIVE0 (MT3).
ANR8328I 002: 4MM volume LIBR05 mounted in drive DRIVE0 (MT3).
ANR8427I CHECKIN LIBVOLUME for volume LIBR05 in library 4MMLIB completed
successfully.
```

4.4.2.2 Auditing the Library

If you insert or remove volumes in a library without using ADSM commands, the physical library inventory can become out of sync with the library information stored in ADSM. To ensure that ADSM's inventory matches the content of the library, use the AUDIT command. The following command causes the contents of the 4MMLIB library to be inventoried.

```
adsm> audit library 4MMLIB
```

Audit checks all volumes in the library, reads their labels, and updates the ADSM library inventory to match that of the physical library.

4.4.2.3 Checking Volumes out of the Library

You may have to remove a volume from a library, for example, when it contains data that you may want to vault. Use the CHECKOUT LIBVOLUME command to eject the volume from the library and inform ADSM of the operation. For example, if you want to eject your previously checked in volume, LIBR05, from the library, issue this command:

```
adsm> checkout libvolume 4MMLIB LIBR05
```

ADSM starts a background process, mounts the volume, checks the label, and then prompts you to remove the volume from the library:

```
ANR8434I CHECKOUT LIBVOLUME: Operation for volume LIBR05 in library 4MMLIB
started as process 45.
ANR8336I Verifying label of 4MM volume LIBR05 in drive DRIVE0 (MT3).
ANR8307I 003: Remove 4MM volume LIBR05 from slot with element number 9 of
library 4MMLIB; issue 'REPLY' along with the request ID when ready.
reply 003
ANR8438I CHECKOUT LIBVOLUME for volume LIBR05 in library 4MMLIB completed
successfully.
```

4.4.3 Autochanger Barcode Support

ADSM supports barcode labels on certain SCSI-attached libraries. The barcode support enables you to improve both the speed of certain commands and the general reliability of tape management procedures. ADSM supports barcode labels with a maximum of six characters. Four ADSM commands or utilities take advantage of barcode support:

- DSMLABEL
- CHECKIN LIBVOLUME
- AUDIT LIBRARY
- CHECKOUT LIBVOLUME

The DSMLABEL utility, with the -barcode option, can use the first six characters of the barcode label as the ADSM volume label. This speeds up labeling operations because ADSM does not have to prompt the user for a label. It also ensures consistency of external and internal labels by eliminating possible user errors.

The other three ADSM commands all have the checklabel option, which you can use to eliminate the label check that is performed by default by all three commands. For example, this command:

```
adsm> checkin libvolume 4MMLIB LIBR05 STAT=SCR checklabel=no
```

causes ADSM to skip the tape mount and label verification that is typically performed during checkin. ADSM trusts the label read by the barcode reader in the library.

4.4.4 Removable File Media

When using removable file media, label the volume by issuing the LABEL command from an NT command prompt or using the Windows NT file manager, label disk option. The label can be up to eight characters long, must not contain embedded blanks, and must be valid when used as a file name on the media.

4.4.5 Volume History File

ADSM maintains a history of all sequential volumes it has used inside its database. You can copy this information to one or more volume history files. The volume history file name is defined in DSMSERV.OPT in the ADSM server directory. To display the volume history file name, use the QUERY OPTION command:

```
adsm> query option

Server Option     Option Setting       Server Option     Option Setting
----------------- -------------------- ----------------- --------------------
CommTimeOut       60                   IdleTimeOut       15
BufPoolSize       512                  LogPoolSize       128
DateFormat        1 (mm/dd/yyyy)       TimeFormat        1 (hh:mm:ss)
NumberFormat      1 (1,000.00)         MessageFormat     1
Language          AMENG                MaxSessions       25
ExpInterval       1                    ExpQuiet          No
MirrorRead DB     Normal               MirrorRead LOG    Normal
MirrorWrite DB    Sequential           MirrorWrite LOG   Parallel
 VolumeHistory    volhist.out          Devconfig         devcnfg.out
TxnGroupMax       16                   MoveBatchSize     32
MoveSizeThresh    1                    StatusMsgCnt      1
TcpPort           1500                 TCPWindowsize     8192
TCPNoDelay        No                   IPXSocket         8522
NetBiosName       ADSMNT               NetbiosBufferSize 16384
NetbiosSessions   16                   IpxBufferSize     32768
NamedPipeName     \\.\PIPE\ADSMPIPE    LanAdapter        0
CommMethod        TCPIP                CommMethod        NETBIOS
CommMethod        NAMEDPIPE            Message Interval  1
MaxMemory         536870912
```

Our volume history file is VOLHIST.OUT, which is the default name. The volume history file contains details of all sequential volumes used within storage pools, for database backups, and for server exports:

```
*******************************************************************************************
*
*               IBM AdStar Distributed Storage Manager Sequential Volume Usage History
*                                    Updated 10/02/1996 18:17:29
*
*     Operation          Volume   Backup Backup Volume Device                      Volume
*     Date/Time           Type    Series Oper.   Seq   Class Name                 Name
*******************************************************************************************
 1996/10/02 11:05:04  STGNEW          0      0      0 4MM                         LIBR01
 1996/10/02 11:05:04  STGNEW          0      0      0 4MM                         LIBR02
 1996/10/02 13:37:33  STGNEW          0      0      0 8MM                         COPY01
 1996/10/02 17:55:36  BACKUPFULL      1      0      1 ZIP                         ZIP001
* export server filedata=none devclass=FILE scratch=yes
 1996/10/02 18:17:29  EXPORT          0      0      1 FILE         D:\ADSMFILE\44305444.EXP
```

This above example shows three tape volumes, two 4MM and one 8MM, that have been added to storage pools and have the STGNEW volume type. One volume, ZIP001, has been used for a full database backup and has the BACKUPFULL volume type. The volume history file is updated automatically whenever a new sequential volume is used. You can also back up the volume history file, using the BACKUP VOLHISTORY administrator command:

```
adsm> backup volhistory file=volbak.txt
```

As with the device configuration file, the optional file= parameter can be used to specify a different volume history file. If this parameter is not used, it is backed up to the files specified in DSMSERV.OPT.

The volume history file is vital for the recovery of the server database in case of database loss. From the volume history file, you can determine the latest database dump volume to be used in recovering the ADSM database.

4.5 Verifying Storage Pool Operation

Having configured the hardware devices and storage pools and labeled the tape, you can now verify that your configuration works. The following example verifies the primary storage pool hierarchy. Use of copy storage pools is covered in Chapter 6, “Storage Pool Availability” on page 97.

Test the setup by backing up data from a workstation. Start a command line client and issue a selective back up command to back up all files on the c: drive and all subdirectories:

```
dsmc> selective -subdir=yes c:\*
```

This will back up the client data to the BACKUPPOOL disk storage pool. When that pool has filled, the server starts a background process to migrate the data to the next pool, our newly configured BACK_COPY pool, which uses scratch tape volumes in our Aiwa library. To monitor the space used in BACKUPPOOL, issue the QUERY STGPOOL BACKUPPOOL command:

```
adsm> query stgpool backuppool
Storage      Device       Estimated  %Util  %Migr  High   Low  Next
Pool Name    Class Name    Capacity               Mig%  Mig%  Storage
                               (MB)                        Pool
-----------  ----------  ----------  -----  -----  ----  ----  ---------
BACKUPPOOL   DISK              4.0   99.9   99.9    80    60  BACK_TAPE
```

The command output shows us that the storage pool is full (99.9% utilization) and that all of the data is eligible for migration.

Using the administrative command line client with the -consolemode option or the server console, you will see messages such as the following indicating that migration processes are starting and ending:

```
ANR1000I Migration process 4 started for storage pool BACKUPPOOL.
ANR8337I 4MM volume LIBR01 mounted in drive DRIVE0 (MT3).
ANR1340I Scratch volume LIBR01 is now defined in storage pool BACK_TAPE.
ANR1001I Migration process 4 ended for storage pool BACKUPPOOL.
```

Notice from message ANR1340I that a scratch volume, LIBR01, previously checked into the library, has been assigned to storage pool BACK_TAPE.

To check whether a storage pool migration process is active, issue the QUERY PROCESS command:

```
adsm> q process
 Process Process Description  Status
  Number
-------- -------------------- -------------------------------------------------
       4 Migration            Disk Storage Pool BACKUPPOOL, Moved Files: 4,
                               Moved Bytes: 20,480, Unreadable Files: 0,
                               Unreadable Bytes: 0. Current File (bytes):
                               45,056

                               Current output volume: LIBR01.
```

The command output shows us the active process numbers and types. One migration process from BACKUPPOOL is writing to LIBR01.

To verify which volumes are assigned to the storage pool, issue the QUERY VOLUME command:

```
adsm> query volume stgpool=BACK_TAPE

Volume Name              Storage      Device      Estimated  %Util   Volume
                         Pool Name    Class Name   Capacity          Status
                                                       (MB)
------------------------ -----------  ----------  ---------  -----  -------
LIBR01                   BACK_TAPE    4MM         3,814.0    3.2    Filling
```

The command output shows us that one volume is assigned to the storage pool and it is 3.2% full. The status is Filling, so when more data is migrated to this storage pool, the same volume will be reused. When the volume becomes full, the server automatically requests a new scratch volume and assigns it to the storage pool.

Part 2. Server Availability

An ADSM server provides archive, backup, and HSM functions for client workstations. This data held on the server is often critical, and maintaining its availability is important. In the second part of the book, we look at ADSM server availability and the server functions to ensure it. In the chapters in Part 2 we provide examples of performing the following ADSM server tasks:

- Database backup and recovery
- Storage pool backup and recovery
- Complete server recovery

In these examples we use the command line administrative client and/or the server console. Most tasks can also be performed through the GUI administrative client; however, we used the command line client to better illustrate the underlying concepts.

Chapter 5. Database Availability

The server database and recovery log are the most critical components of an ADSM server. Loss of either component results in the loss of the server and all its managed client workstation data. Planning database availability and recovery is a key element in an ADSM implementation.

In this chapter we discuss the ADSM functions available for protection against loss or damage of the database and recovery log. In the sections that follow we provide examples of:

- Database and recovery log mirroring
- Database backup and recovery

5.1 Database and Recovery Log Mirroring

ADSM provides database and recovery log volume mirroring capabilities for server availability. ADSM mirroring is established at the volume level. It is flexible and is activated and deactivated under ADSM control. ADSM mirroring provides enhanced server availability, protecting against local media and hardware failures. Without mirroring, loss of a database or recovery log volume would cause the server to fail. With mirroring, loss of a mirror copy because of hardware failure does not impact the operation of the server.

Although database and recovery log mirroring provides a high degree of protection, it does have a cost. Sufficient extra disk space must be available for the mirrored copies. If you have to choose between the database or the recovery log, mirroring the recovery log is the sensible choice. The database can be recovered from database backups. This recovery can be more current if the recovery log is still available. The combination of mirroring the recovery log and performing regular database backups is a good compromise configuration. It provides a high degree of resilience without needing an excessive amount of additional disk space.

Figure 47 on page 78 shows an example of extension and mirroring. The concepts apply to both the database and recovery log. We have three disk volumes: Disk1, Disk2 and Disk3. The database can be mirrored across the disk volumes, for example, volume V1M1 is mirrored to both V1M2 on Disk2 and V1M3 on Disk3. The database can also be extended if the original space is no longer sufficient. In our example the database is made up of a small V1M1 and a larger V2M1. Volume V2M1 can then be mirrored to V2M2 and V2M3.

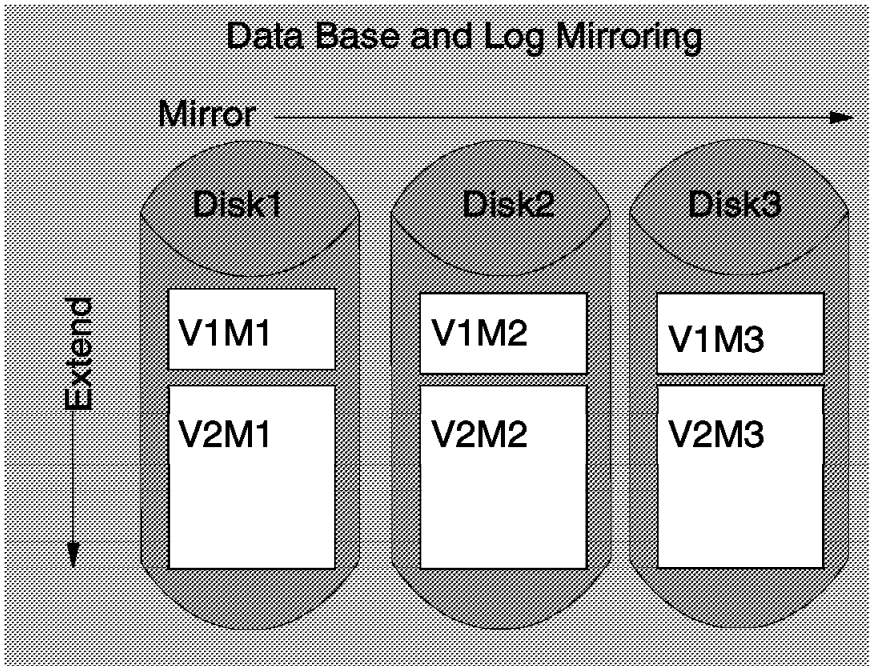


Figure 47. Database and Log Extension and Mirroring

We now present an example of setting up recovery log volume mirroring with two mirrored copies. Before you start mirroring the recovery log, you must decide on which volumes to place the mirrored copies. To determine where your recovery logs are located and their size, issue the QUERY LOGVOL command:

```
adsm> query logvol f=d
Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\LOG1.DSM
         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
Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\LOG2.DSM
         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
```

The recovery log volumes are located on the C: drive. For maximum protection create a new directory, in this example, ADSMMIR, on a different drive, drive D:. Using the DSMFMT disk volume formatting utility, create two new recovery log volumes. Change to the D:\ADSMMIR directory and enter the following commands:

```
D:\ADSMMIR> c:\win32app\ibm\adsm\utils\dsmfmt -log log1.dsm 25
D:\ADSMMIR> c:\win32app\ibm\adsm\utils\dsmfmt -log log2.dsm 25
```

Two 24 MB volumes have now been formatted, and you can define them to ADSM by using the DEFINE LOGCOPY commands:

```
adsm> define logcopy log1.dsm  d:\adsmmir\log1.dsm
adsm> define logcopy log2.dsm  d:\adsmmir\log2.dsm
```

These commands create mirrored copies of the LOG1.DSM and LOG2.DSM volumes with the newly formatted mirrored copies in the D:\ADSMMIR directory. The command starts a background process that you monitor from the console or administrative client:

```
ANR2262I Recovery log volume copy D:\ADSMMIR\LOG1.DSM defined; synchronization
process started (process ID 100).
ANR0221I Synchronization of recovery log volume D:\ADSMMIR\LOG1.DSM started as
process 100.
ANR0235I Synchronization complete of recovery log volume D:\ADSMMIR\LOG1.DSM.
 ....
```

From these messages you can verify that the recovery log has been mirrored and that the two mirrored copies are now synchronized. Use the QUERY LOGVOL command again to verify that the recovery log has been synchronized:

```
adsm> query logvol f=d

Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\LOG1.DSM
         Copy Status: Sync'd
Volume Name (Copy 2): D:\ADSMMIR\LOG1.DSM
         Copy Status: Sync'd
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 24
Allocated Space (MB): 24
     Free Space (MB): 0
Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\LOG2.DSM
         Copy Status: Sync'd
Volume Name (Copy 2): D:\ADSMMIR\LOG2.DSM
         Copy Status: Sync'd
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 24
Allocated Space (MB): 24
     Free Space (MB): 0
```

To verify that the database has been mirrored, ensure that Volume Name (Copy 2) field has a volume name and the Copy Status field shows *Sync'd*.

To test that your mirroring works, you can stop the server and rename the C:\win32app\ibm\adsm\server\log1.dsm recovery log volume, simulating a hardware failure. When you restart the server, you get these messages:

```
ANR0990I ADSM server restart-recovery in progress.
ANR0215W Recovery log volume C:\WIN32APP\IBM\ADSM\SERVER\LOG1.DSM is in the
offline state - VARY ON required.
 ...
```

The server detects that the recovery log volume is missing, issues warning messages, and continues the server startup, using the mirrored copy of the volume. If you then rename the temporary copy back to C:\win32app\ibm\adsm\server\log1.dsm and issue the VARY ON command to make the volume available to the server, you will see the following messages:

```
adsm> vary on log1.dsm
ANR0221I Synchronization of recovery log volume ...
           ...C:\WIN32APP\IBM\ADSM\SERVER\LOG1.DSM started as process 2.
ANR0235I Synchronization complete for recovery log volume ...
           ...C:\WIN32APP\IBM\ADSM\SERVER\LOG1.DSM.
```

You can see that the server started a background process to resynchronize the log volume and that the process completed successfully.

5.2 Database Backup and Recovery

The database is a critical component of the server. The server will not start if it is unavailable. In addition to mirroring the database, you can back it up online. In this section we describe the tasks involved in backing up and recovering the database.

5.2.1 Backing Up the Database

You back up a database while the server is running, using the BACKUP DB command. Backups can be either full or incremental. The output device is specified with a device class. In this example we use our manual 8 mm drive for the backup, first for a full and then an incremental backup.

To perform a full database backup with the output of the backup sent to the 8 mm drive, enter the following command:

```
adsm> BACKUP DB devclass=8MM type=full
```

This starts a full database backup, which you can monitor either at the server console, from an administrative client running in console mode, or by browsing the server activity log:

```
ANR2280I Full database backup started as process 2.
ANR8326I 001: Mount 8MM volume SCRTCH R/W in drive DRIVE1 (MT1.0.0.1) of
library 8MMLIB within 60 minutes.
ANR8328I 001: 8MM volume BACK01 mounted in drive DRIVE1 (MT1.0.0.1).
ANR1360I Output volume BACK01 opened (sequence number 1).
ANR4554I Backed up 32 of 1499 database pages.
ANR4554I Backed up 208 of 1499 database pages.
ANR4554I Backed up 384 of 1499 database pages.
ANR4554I Backed up 560 of 1499 database pages.
ANR4554I Backed up 736 of 1499 database pages.
ANR4554I Backed up 912 of 1499 database pages.
ANR4554I Backed up 1088 of 1499 database pages.
ANR4554I Backed up 1264 of 1499 database pages.
ANR4554I Backed up 1440 of 1499 database pages.
ANR1361I Output volume BACK01 closed.
ANR4550I Full database backup (process 2) complete, 1499 pages copied.
ANR8336I Verifying label of 8MM volume BACK01 in drive DRIVE1 (MT1.0.0.1)
ANR8468I 8MM volume BACK01 dismounted from drive DRIVE1 (MT1.0.0.1) in library
8MMLIB.
```

The database backup process by default uses scratch tapes. If you want to use a specific volume, you can specify a volume name as a parameter with the BACKUP DB command.

At some point later, after client activity has added more data to the storage pools and updated the database, you can back up the database again. This time, rather than doing another full backup, you can do an incremental backup. Again, use the BACKUP DB command, but this time with the type=incremental option:

```
adsm> BACKUP DB devclass=8MM type=incr
```

An incremental backup backs up only those database pages that have changed since the previous backup. You can check for successful completion of the backup process:

```
ANR2281I Incremental database backup started as process 3.
ANR8326I 002: Mount 8MM volume SCRTCH R/W in drive DRIVE1 (MT1.0.0.1) of
library 8MMLIB within 60 minutes.
ANR8328I 002: 8MM volume BACK02 mounted in drive DRIVE1 (MT1.0.0.1).
ANR1360I Output volume BACK02 opened (sequence number 1).
ANR4554I Backed up 32 of 35 database pages.
ANR1361I Output volume BACK02 closed.
ANR4551I Incremental database backup (process 3) complete, 35 pages copied
ANR8336I Verifying label of 8MM volume BACK02 in drive DRIVE1 (MT1.0.0.1)
ANR8468I 8MM volume BACK02 dismounted from drive DRIVE1 (MT1.0.0.1) in library
8MMLIB.
```

Full and incremental database backups are logically grouped together as a *backup series*, which consists of a full backup and up to 32 incremental backups. When 32 incremental backups have been performed, a new backup series must be started with a new full backup. Details of database backups, the backup series, and the volumes used are also kept in the server volume history file. To view the file, issue the administrator QUERY VOLHISTORY command or look at the backup volume history file, which contains a summary of the volume history:

```
*******************************************************************************
*
*               IBM AdStar Distributed Storage Manager Sequential Volume Usage History
*                                    Updated 10/21/1996 11:03:06
*
*     Operation         Volume   Backup Backup Volume Device                    Volume
*     Date/Time          Type    Series Oper.   Seq   Class Name               Name
*******************************************************************************
 1996/10/10 11:24:23  STGNEW          0      0      0 4MM                       LIBR01
 1996/10/11 09:22:07  STGNEW          0      0      0 4MM                       LIBR02
 1996/10/21 10:45:32  BACKUPFULL      1      0      1 8MM                       BACK01
 1996/10/21 11:01:10  BACKUPINCR      1      1      1 8MM                       BACK02
```

From this you can see the two database backup volumes, BACK01 for the full and BACK02 for the incremental backup.

You can decide how your database backups are performed. You could simply perform full backups on a regular basis or do weekly full backups with daily incremental backups in between. The only limitation is that you cannot do more than 32 incremental backups between full backups.

Backup Volumes

Database backups require a new volume for every backup, either full or incremental. Backups cannot be appended to existing database backup or storage pool volumes. In view of this, care should be taken in selecting the device type for database backups. We suggest that you use an appropriate device for the quantity of data expected from the database backups.

In our example we use the same device class for full and incremental backups. If you want, you can use different device classes.

5.2.2 Database Recovery Modes

ADSM can perform full or incremental backups of the database while the server is operational. Two modes of server operation determine how the database backups are used during a recovery:

- Normal mode
- Roll forward mode

When the server is running in roll forward mode, you can define database backup triggers.

5.2.2.1 Normal Mode

In normal mode the database can be recovered only up to the most recent database backup. This is known as *point in time recovery*. To perform a point-in-time recovery, restore the most recent full database backup and then restore subsequent incremental backups until the required recovery point is reached. This is a stand-alone, full database recovery. As long as the database backup volumes are available, the database can be recovered, even if the server has to be reinstalled. However, any client workstation data sent to the server after the last database backup will be lost. This is the normal mode of operation for the server.

5.2.2.2 Roll Forward Mode

In roll forward mode the database can be recovered up to the last completed client workstation transaction, as long as the server recovery log is still available. Roll forward mode uses the recovery log to record all changes made to the database since the last database backup, either full or incremental. These database transactions are accumulated in the recovery

log until the next database backup takes place, at which point they are discarded and the process starts again. The transactions recorded in the recovery log are used to roll forward the database after it has been restored.

When the server is installed, the database is set to normal mode. To activate roll forward mode, issue the following administrator command:

```
adsm> set logmode rollforward
```

The additional records in the recovery log can cause a significant increase in the size of the log, and you must take this increase in size into consideration when implementing roll forward recovery.

Note

You can recover a database in roll forward mode only if the server recovery log is still available. If it has been lost, you must perform a point-in-time recovery. If you set roll forward mode, both types of recovery can be performed.

5.2.2.3 Database Backup Triggers

In roll forward mode, database changes are saved to the recovery log. Thus the recovery log must be of sufficient size to hold the database updates accumulated since the last database backup. In a situation where client activity is unusually high, the recovery log could run out of space before the next scheduled database backup is performed. If the recovery log runs out of space, the server stops. To prevent the recovery log from running out of space, use a database backup trigger.

A database backup trigger defines a maximum utilization for the recovery log. When that utilization is reached, a database backup is triggered to free space in the recovery log. Triggers can be defined to perform full backups only, or a combination of full and incremental. This example shows a trigger defined so that a backup takes place if the recovery log utilization reaches 75%:

```
adsm> define dbbackuptrigger logfullpct=75 devc=8MM numincremental=0
```

The triggered database backups will use the 8MM device class and will always be full backups (this is controlled by specifying numincremental=0).

If the recovery log utilization reaches 75%, a backup is triggered and the following will be seen on the server console or in the activity log:

```
ANR4552I Full database backup triggered; started as
 process 17.
ANR8326I 001: Mount 8MM volume SCRTCH R/W in drive DRIVE1
 (MT1.0.0.1) of library 8MMLIB within 60 minutes.
ANR8328I 002: 8MM volume BACK03 mounted in drive DRIVE1
 (MT1.0.0.1).
ANR1360I Output volume BACK03 opened (sequence number 1).
ANR4554I Backed up 32 of 815 database pages.
ANR4554I Backed up 544 of 815 database pages.
ANR1361I Output volume BACK03 closed.
ANR4550I Full database backup (process 17) complete, 815
 pages copied.
ANR8336I Verifying label of 8MM volume BACK03 in drive
 DRIVE1 (MT1.0.0.1).
ANR8468I 8MM volume BACK03 dismounted from drive DRIVE1
 (MT1.0.0.1) in library 8MMLIB.
```

This database backup deletes the accumulated transaction records in the recovery log, reducing its utilization.

Caution

Database backup triggers are designed as a “safety valve” to prevent the recovery log from running out of space while the server is running in roll forward mode. Triggers are not intended as alternatives to regular database backups, which should be scheduled on a regular basis. Use backup triggers only when the recovery log grows faster than anticipated.

5.2.3 Database Recovery

Use the DSMSERV RESTORE DB utility to restore the server database. DSMSERV RESTORE DB is a stand-alone utility, and the server must not be operational when it is run.

To perform the restore, you must have a full database backup, and any incremental backups taken between the last full backup and the point in time to which you are recovering. In addition, the following information is required:

- The database and recovery log sizes
- DSMSERV.OPT
- Server disk configuration file, DSMSERV.DSK
- A backup of the volume history file
- A backup of the device configuration file

To obtain this information we set up the following ADSM macro:

```
backup volhist
backup devconfig
query db f=d
query dbv f=d
query log f=d
query logv f=d
query volhist type=dbb
query option
query sta
query dbb f=d
```

To run the macro, save the commands in a macro file and start an administrative session in batch mode:

```
dsmadmc -id=admin -pass=passw macro MACRO_NAME  > OUT_FILE_NAME
```

This command will run the macro in MACRO_NAME and redirect the output to OUT_FILE_NAME. We suggest that you run such a macro on a regular basis and save the output report in a secure place. For an example of the output of this macro, see A.3, “Sample SERVER.REP” on page 131.

Note

The macro must be run in batch mode for the output to be formatted correctly. Running a macro and redirecting the output to file from an interactive administration session will result in garbled output.

In the sections that follow we illustrate three different database recoveries:

1. Point-in-time database recovery
2. Roll forward recovery (full database)
3. Roll forward recovery (single database volume)

The first example is a stand-alone recovery suitable for offsite server recovery. The second and third examples are suitable for local server recovery where either the entire database or a single database volume must be recovered.

5.2.3.1 Point-in-Time Database Recovery

In this example a single database volume has been lost, causing the server to go down. To restore the entire database, use the latest full and incremental backups. Before you can restore the database, you must re-create the lost database volume.

First look at the output report produced from the ADSM macro that you ran and locate the section containing the detailed report of the server's database volumes. From this report you can determine the volume names from Volume Name (Copy 1) field and their sizes from the Allocated Space (MB) field:

```
ANS5101I Server command: 'q dbv f=d'
Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\DB1.DSM
         Copy Status: Sync'd
Volume Name (Copy 2):
         Copy Status: Undefined
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 14
Allocated Space (MB): 14
     Free Space (MB): 0
Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\DB2.DSM
         Copy Status: Sync'd
Volume Name (Copy 2):
         Copy Status: Undefined
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 14
Allocated Space (MB): 14
```

Before you start the database recovery, you must define new database volumes to replace the lost volumes. You are restoring the complete database because you lost a single database volume. The other database volumes and the recovery log are still intact. To perform this recovery, format a replacement volume with the same name and size as the lost volume.

In this example, one of the volumes, DB1.DSM, has been lost. You re-create the lost volume and then restore the database. At a command prompt, change to the directory where the database resided, C:\win32app\ibm\adsm\server. In that directory execute the ADSM disk volume formatting utility, DSMFMT. As parameters, specify that you want to format a database volume (-db); the name of the volume is db1.dsm, and the size is 15 MB:

```
C:\win32app\ibm\adsm\server>DSMFMT -DB DB1.DSM 15
```

You have created and formatted a server database volume of 14 MB.

Volume Sizes

Database volumes must be allocated in multiples of 4 MB plus 1 MB for formatting information.

You are now ready to start the database restore, using the DSMSERV RESTORE DB command. To do a full, point-in time recovery, the following are required:

- DSMSERV.OPT
- DSMSERV.DSK
- Volume history backup file
- Device configuration file

First, browse the latest backup copy of the volume history backup file to determine the latest database backup series:

```
*******************************************************************************************
*
*               IBM AdStar Distributed Storage Manager Sequential Volume Usage History
*                                    Updated 10/23/1996 09:07:15
*
*     Operation          Volume   Backup Backup Volume Device                      Volume
*     Date/Time           Type    Series Oper.   Seq   Class Name                 Name
*******************************************************************************************
 1996/10/10 11:24:23  STGNEW          0      0      0 4MM                         LIBR01
 1996/10/11 09:22:07  STGNEW          0      0      0 4MM                         LIBR02
 1996/10/21 10:45:32  BACKUPFULL      1      0      1 8MM                         BACK01
 1996/10/21 11:01:10  BACKUPINCR      1      1      1 8MM                         BACK02
```

From this you can see that the latest backup series, Series 1, consists of two volumes, a full backup (BACK01) and an incremental (BACK02), and the device class used for these is 8MM. This is the backup series you are going to restore.

Second, you need the DSMSERV.OPT and device configuration files. The server options file has an option, *DevConfig*, which points to the backup device configuration file. The DSMSERV RESTORE DB command must mount the backup volumes, BACK01 and BACK02, to perform the restore. You tell it the device class to use, in this example, 8MM. With this information, the restore command then uses DSMSERV.OPT to locate the backup device configuration file. Once it locates the file, it uses the device configuration and associated library and drive definitions to mount the backup volumes.

The backup series consists of two volumes. To illustrate the steps involved in restoring with multiple volumes, we will run the restore process in two stages: First the full backup is restored, and then the incremental backup is restored and the changes committed. Start the first restore by specifying

the device class, backup volume name, and the commit=no option (Figure 48 on page 89).

```
C:\win32app\ibm\adsm\server>dsmserv restore db devc=8MM vol=back01 commit=no

ANR0900I Processing options file C:\win32app\ibm\adsm\server\dsmserv.opt.

ADSTAR Distributed Storage Manager for Windows NT
Version 2, Release 1, Level 0.12/0.12

Licensed Materials - Property of IBM

5639-A09 (C) Copyright IBM Corporation 1990, 1996. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 22:06:54 on Nov 18 1996.
ANR0200I Recovery log assigned capacity is 48 megabytes.
ANR0201I Database assigned capacity is 28 megabytes.
ANR8324I 8MM volume BACK01 is expected to be mounted (R/O).
ANR4621I Database backup device class 8MM.
ANR4622I   Volume 1: BACK01.
ANR4632I Starting point-in-time database restore (no commit).
ANR8326I 001: Mount 8MM volume BACK01 R/O in drive DRIVE1 (MT1.0.0.1) of library
8MMLIB within 60 minutes.
ANR8335I 001: Verifying label of 8MM volume BACK01 in drive DRIVE1 (MT1.0.0.1).
ANR8328I 001: 8MM volume BACK01 mounted in drive DRIVE1 (MT1.0.0.1).
ANR1363I Input volume BACK01 opened (sequence number 1).
ANR4646I Database capacity required for restore is 28 megabytes.
ANR4638I Restore of backup series 1 operation 0 in progress.
ANR4639I Restored 192 of 1692 database pages.
ANR4639I Restored 400 of 1692 database pages.
ANR4639I Restored 608 of 1692 database pages.
ANR4639I Restored 816 of 1692 database pages.
ANR4639I Restored 1024 of 1692 database pages.
ANR4639I Restored 1232 of 1692 database pages.
ANR4639I Restored 1440 of 1692 database pages.
ANR4639I Restored 1648 of 1692 database pages.
ANR1364I Input volume BACK01 closed.
ANR4640I Restored 1692 pages from backup series 1 operation 0.
ANR4633I Point-in-time database restore (no commit) complete.
ANR8468I 8MM volume BACK01 dismounted from drive DRIVE1 (MT1.0.0.1) in library
8MMLIB.
```

Figure 48. Point-in-Time Database Restore (First Volume): Restoring Full Backup

The full database backup is restored to the server database volumes, but the changes are not committed. At this point the server cannot be restarted. You must complete the second restore of the incremental backup, with the commit=yes option (Figure 49).

```
C:\win32app\ibm\adsm\server>dsmserv restore db devc=8MM vol=back02 commit=yes

ANR0900I Processing options file C:\win32app\ibm\adsm\server\dsmserv.opt.

ADSTAR Distributed Storage Manager for Windows NT
Version 2, Release 1, Level 0.12/0.12

Licensed Materials - Property of IBM
```

Figure 49 (Part 1 of 2). Point-in-Time Database Recovery (Second Volume): Restoring Incremental Backup and Committing Changes

```
5639-A09 (C) Copyright IBM Corporation 1990, 1996. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 22:06:54 on Nov 18 1996.
ANR0200I Recovery log assigned capacity is 48 megabytes.
ANR0201I Database assigned capacity is 28 megabytes.
ANR8324I 8MM volume BACK02 is expected to be mounted (R/O).
ANR4621I Database backup device class 8MM.
ANR4622I   Volume 1: BACK02.
ANR4630I Starting point-in-time database restore (commit).
ANR8326I 001: Mount 8MM volume BACK02 R/O in drive DRIVE1 (MT1.0.0.1) of library
8MMLIB within 60 minutes.
ANR0300I Recovery log format started; assigned capacity 48 megabytes.
ANR0301I Recovery log format in progress; 4 megabytes of 48.
ANR0301I Recovery log format in progress; 8 megabytes of 48.
ANR0301I Recovery log format in progress; 12 megabytes of 48.
ANR0301I Recovery log format in progress; 16 megabytes of 48.
ANR0301I Recovery log format in progress; 20 megabytes of 48.
ANR0301I Recovery log format in progress; 24 megabytes of 48.
ANR0301I Recovery log format in progress; 28 megabytes of 48.
ANR0301I Recovery log format in progress; 32 megabytes of 48.
ANR0301I Recovery log format in progress; 36 megabytes of 48.
ANR0301I Recovery log format in progress; 40 megabytes of 48.
ANR8335I 001: Verifying label of 8MM volume BACK02 in drive DRIVE1 (MT1.0.0.1).
ANR0301I Recovery log format in progress; 44 megabytes of 48.
ANR8328I 001: 8MM volume BACK02 mounted in drive DRIVE1 (MT1.0.0.1).
ANR1363I Input volume BACK02 opened (sequence number 1).
ANR4638I Restore of backup series 1 operation 1 in progress.
ANR0301I Recovery log format in progress; 48 megabytes of 48.
ANR0302I Recovery log formatting took 53000 milliseconds.
ANR0303I   Format rate:         231.8  pages/second.
ANR0304I   Page service time:     4.3  ms.
ANR0305I Recovery log format complete.
ANR4640I Restored 57 pages from backup series 1 operation 1.
ANR0306I Recovery log volume mount in progress.
ANR4641I Sequential media log redo pass in progress.
ANR4642I Sequential media log undo pass in progress.
ANR1364I Input volume BACK02 closed.
ANR4644I A full backup will be required for the next database backup operation.
ANR4631I Point-in-time database restore (commit) complete, restore date
11/21/1996 18:41:41.
ANR8468I 8MM volume BACK02 dismounted from drive DRIVE1 (MT1.0.0.1) in library
8MMLIB.
```

Figure 49 (Part 2 of 2). Point-in-Time Database Recovery (Second Volume): Restoring Incremental Backup and Committing Changes

Running the incremental backup restore with commit=yes formats the server recovery log, restores the incremental backup to the previously restored full backup, and commits the changes. When the restore is complete, you can restart the server.

Note

The above example is of a point-in-time recovery specifying volume names. You can perform this database restore in a single step where the volumes are requested as needed. For an example see Chapter 7, “Server Recovery Example” on page 107.

The above point-in-time restore example allows you to recover the database without reinstalling the server. It is a straightforward database recovery process that you can use for local recovery where a database volume has

been lost but the other server database and recovery log volumes are still available.

Restoring the database with this method may well produce inconsistencies between the restored database and the storage pools. Any changes to the database following the last incremental backup will have been lost. You will have to correct these inconsistencies by auditing the storage pool volumes. For local recovery where this loss of data is unacceptable, set the server to roll forward mode and recover with the aid of the server recovery log.

5.2.3.2 Roll Forward Recovery (Full Database)

Roll forward recovery of the database enables recovery to the last committed client workstation transaction. This mode of recovery requires that the server recovery log be available at the time of recovery (5.2.2, “Database Recovery Modes” on page 83). If the recovery log has also been lost, a point-in-time recovery must be performed as discussed in the previous section. To enable roll forward recovery, you must set roll forward mode at the server console or from an administrator client:

```
adsm> set logmode rollforward
ANR2294I Log mode set to ROLLFORWARD.
```

Roll forward mode will not take effect until a new database backup series is started with a full backup. Once the new backup series has been started, the server will start recording client transactions in the recovery log.

The following example is based on a new backup series, this time performed using the Zip drive and the ZIP device class. This new database backup series, Series 4, consists of one full backup and one incremental backup and was started after setting the server to roll forward mode. You can see the new backup series from the volume history file:

```
*******************************************************************************
*
*               IBM AdStar Distributed Storage Manager Sequential Volume Usage History
*                                    Updated 10/25/1996 10:12:20
*
*     Operation          Volume   Backup Backup Volume Device                    Volume
*     Date/Time           Type    Series Oper.   Seq   Class Name                Name
*******************************************************************************
 1996/10/10 11:24:23  STGNEW          0      0      0 4MM                       LIBR01
 1996/10/11 09:22:07  STGNEW          0      0      0 4MM                       LIBR02
 1996/10/21 10:45:32  BACKUPFULL      1      0      1 8MM                       BACK01
 1996/10/21 11:01:10  BACKUPINCR      1      1      1 8MM                       BACK02
 1996/10/20 08:38:15  BACKUPFULL      4      0      1 ZIP                       ZIP001
 1996/10/21 14:51:20  BACKUPINCR      4      1      1 ZIP                       ZIP002
```

To perform a roll forward database restore, the following must be available:

- Server recovery log volumes
- DSMSERV.OPT
- DSMSERV.DSK
- Volume history backup file
- Device configuration file

The database volumes that have been lost must first be re-created with the same name and size; use the DSMFMT utility. A full recovery in this mode is more automated than a point-in-time recovery. You do not have to specify any options with the DSMSERV RESTORE DB command. The process uses the VolumeHistory option of DSMSERV.OPT to locate the volume history backup file. From that, the process determines the volumes in the latest backup series and their associated device classes. With that information, it can start the restore as before, using the information in the device configuration backup file to mount the volumes (Figure 50).

```
C:\win32app\ibm\adsm\server>dsmserv restore db
ANR0900I Processing options file C:\win32app\ibm\adsm\server\dsmserv.opt.

ADSTAR Distributed Storage Manager for Windows NT
Version 2, Release 1, Level 0.12/0.12

Licensed Materials - Property of IBM

5639-A09 (C) Copyright IBM Corporation 1990, 1996. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 22:06:54 on Nov 18 1996.
ANR0200I Recovery log assigned capacity is 48 megabytes.
ANR0201I Database assigned capacity is 28 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR8324I REMOVABLEFILE volume ZIP001 is expected to be mounted (R/O).
ANR8324I REMOVABLEFILE volume ZIP002 is expected to be mounted (R/O).
ANR4620I Database backup series 4 operation 0 device class ZIP.
ANR4622I   Volume 1: ZIP001.
ANR4620I Database backup series 4 operation 1 device class ZIP.
ANR4622I   Volume 1: ZIP002.
ANR4636I Starting roll-forward database restore.
ANR8326I 001: Mount REMOVABLEFILE volume ZIP001 R/O in drive ZIP_DRIVE (F:) of
library ZIPLIB within 60 minutes.
ANR8335I 001: Verifying label of REMOVABLEFILE volume ZIP001 in drive ZIP_DRIVE
(F:).

ANR8328I 001: REMOVABLEFILE volume ZIP001 mounted in drive ZIP_DRIVE (F:).
ANR8340I REMOVABLEFILE volume ZIP001 mounted.
ANR1363I Input volume ZIP001 opened (sequence number 1).
ANR4638I Restore of backup series 4 operation 0 in progress.
ANR4639I Restored 208 of 1866 database pages.
ANR4639I Restored 432 of 1866 database pages.
ANR4639I Restored 656 of 1866 database pages.
ANR4639I Restored 880 of 1866 database pages.
ANR4639I Restored 1104 of 1866 database pages.
ANR4639I Restored 1328 of 1866 database pages.
ANR4639I Restored 1552 of 1866 database pages.
ANR4639I Restored 1776 of 1866 database pages.
ANR1364I Input volume ZIP001 closed.
ANR4640I Restored 1866 pages from backup series 4 operation 0.
ANR8317I The volume in REMOVABLEFILE drive ZIP_DRIVE (F:) must be manually
ejected.
```

Figure 50 (Part 1 of 2). Roll Forward Full Database Restore

```
ANR8468I REMOVABLEFILE volume ZIP001 dismounted from drive ZIP_DRIVE (F:) in
library ZIPLIB.
ANR8326I 002: Mount REMOVABLEFILE volume ZIP002 R/O in drive ZIP_DRIVE (F:) of
library ZIPLIB within 60 minutes.
ANR8335I 002: Verifying label of REMOVABLEFILE volume ZIP002 in drive ZIP_DRIVE
(F:).

ANR8328I 002: REMOVABLEFILE volume ZIP002 mounted in drive ZIP_DRIVE (F:).
ANR8340I REMOVABLEFILE volume ZIP002 mounted.
ANR1363I Input volume ZIP002 opened (sequence number 1).
ANR4638I Restore of backup series 4 operation 1 in progress.
ANR4640I Restored 151 pages from backup series 57 operation 1.
ANR4641I Sequential media log redo pass in progress.
ANR1364I Input volume ZIP002 closed.
ANR0354I Recovery log redo pass in progress.
ANR8317I The volume in REMOVABLEFILE drive ZIP_DRIVE (F:) must be manually
ejected.
ANR4643I Processed 4096 log records.
ANR4637I Roll-forward database restore complete.
ANR8468I REMOVABLEFILE volume ZIP002 dismounted from drive ZIP_DRIVE (F:) in
library ZIPLIB.
```

Figure 50 (Part 2 of 2). Roll Forward Full Database Restore

The restore process prompts for the backup series volumes to be mounted in sequence and restores the database to the latest backup in the series. It then runs a “redo pass” of the recovery log, rolling forward transactions that took place after the last database backup. After the redo pass has completed, the server can be restarted. Its recovered database will be consistent up to the last client transaction.

5.2.3.3 Roll Forward Recovery (Single Database Volume)

The previous two examples showed how the entire database can be recovered, with either a point-in-time or roll forward recovery. The database often consists of multiple database volumes. With roll forward mode set, a single database volume as opposed to the entire database can also be recovered. As with the previous example, you must re-create the volume to be recovered with the same name and size; use the DSMFMT utility. Then you can restore the database contents for that individual volume, using the DSMSERV RESTORE DB command, this time specifying the volume name as a parameter (Figure 51).

```
C:\win32app\ibm\adsm\server>dsmserv restore db dbvol=db2.dsm
ANR0900I Processing options file C:\win32app\ibm\adsm\server\dsmserv.opt.

ADSTAR Distributed Storage Manager for Windows NT
Version 2, Release 1, Level 0.12/0.12

Licensed Materials - Property of IBM

5639-A09 (C) Copyright IBM Corporation 1990, 1996. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 22:06:54 on Nov 18 1996.
ANR0200I Recovery log assigned capacity is 48 megabytes.
```

Figure 51 (Part 1 of 2). Roll Forward Single Database Volume Restore

```
ANR0201I Database assigned capacity is 28 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR8324I REMOVABLEFILE volume ZIP001 is expected to be mounted (R/O).
ANR8324I REMOVABLEFILE volume ZIP002 is expected to be mounted (R/O).
ANR4620I Database backup series 4 operation 0 device class ZIP.
ANR4622I   Volume 1: ZIP001.
ANR4620I Database backup series 4 operation 1 device class ZIP.
ANR4622I   Volume 1: ZIP002.
ANR4636I Starting roll-forward database restore.
ANR8326I 001: Mount REMOVABLEFILE volume ZIP001 R/O in drive ZIP_DRIVE (F:) of
library ZIPLIB within 60 minutes.
ANR8335I 001: Verifying label of REMOVABLEFILE volume ZIP001 in drive ZIP_DRIVE
(F:).

ANR8328I 001: REMOVABLEFILE volume ZIP001 mounted in drive ZIP_DRIVE (F:).
ANR8340I REMOVABLEFILE volume ZIP001 mounted.
ANR1363I Input volume ZIP001 opened (sequence number 1).
ANR4638I Restore of backup series 4 operation 0 in progress.
ANR1364I Input volume ZIP001 closed.
ANR8317I The volume in REMOVABLEFILE drive ZIP_DRIVE (F:) must be manually
ejected.
ANR4640I Restored 3 pages from backup series 4 operation 0.
ANR8468I REMOVABLEFILE volume ZIP001 dismounted from drive ZIP_DRIVE (F:) in
library ZIPLIB.
ANR8326I 002: Mount REMOVABLEFILE volume ZIP002 R/O in drive ZIP_DRIVE (F:) of
library ZIPLIB within 60 minutes.
ANR8335I 002: Verifying label of REMOVABLEFILE volume ZIP002 in drive ZIP_DRIVE
(F:).

ANR8328I 002: REMOVABLEFILE volume ZIP002 mounted in drive ZIP_DRIVE (F:).
ANR8340I REMOVABLEFILE volume ZIP002 mounted.
ANR1363I Input volume ZIP002 opened (sequence number 1).
ANR4638I Restore of backup series 4 operation 1 in progress.
ANR4640I Restored 0 pages from backup series 57 operation 1.
ANR4641I Sequential media log redo pass in progress.
ANR1364I Input volume ZIP002 closed.
ANR0354I Recovery log redo pass in progress.
ANR8317I The volume in REMOVABLEFILE drive ZIP_DRIVE (F:) must be manually
ejected.
ANR4643I Processed 4096 log records.
ANR4637I Roll-forward database restore complete.
ANR8468I REMOVABLEFILE volume ZIP002 dismounted from drive ZIP_DRIVE (F:) in
library ZIPLIB.
```

Figure 51 (Part 2 of 2). Roll Forward Single Database Volume Restore

With a roll forward single database volume restore, only those database pages required for the specified volume are restored. For databases with multiple volumes, the ability to restore individual volumes can speed the recovery process.

5.2.4 Managing Database Backups

It is important to perform database backups on a regular basis by using an administrative schedule (see 7.1.2, “Server Scheduling” on page 109). It is best to back up the database after such operations as scheduled workstation backups and scheduled server storage pool backups.

Running backups on a daily basis will accumulate a large number of database backup volumes over time. None of these volumes can be reused while the server is keeping them as database backup volumes. We suggest that you maintain a certain number of versions of database backups,

allowing old backups to be deleted, by deleting database backups from the server volume history.

In this example we are running daily full database backups but want to maintain only the last three backup versions. Use the DELETE VOLHISTORY TYPE=DBBACKUP command:

```
adsm> delete volhistory type=dbbackup todate=-3
```

This command deletes all database backup versions older than three days from the volume history and returns the backup volumes to scratch status. These scratch volumes can then be reused for new database backups or as storage pool volumes. This command can be scheduled on a regular basis, in conjunction with the daily database backup.

The following example illustrates the process. It involves:

1. Running a database backup
2. Deleting the backup from the volume history
3. Checking the status of the backup volume

First back up the database, using an automated library:

```
adsm> backup db devclass=4MM type=full scratch=yes
```

This backup uses a scratch volume within the automated library. On completion, you can look at the volume history file to confirm this:

```
*******************************************************************************
*
*               IBM AdStar Distributed Storage Manager Sequential Volume Usage History
*                                    Updated 10/25/1996 15:11:17
*
*     Operation          Volume   Backup Backup Volume Device                    Volume
*     Date/Time           Type    Series Oper.   Seq   Class Name                Name
*******************************************************************************
 1996/10/10 11:24:23  STGNEW          0      0      0 4MM                       LIBR01
 1996/10/11 09:22:07  STGNEW          0      0      0 4MM                       LIBR02
 1996/10/21 10:45:32  BACKUPFULL      1      0      1 8MM                       BACK01
 1996/10/21 11:01:10  BACKUPINCR      1      1      1 8MM                       BACK02
 1996/10/20 08:38:15  BACKUPFULL      4      0      1 ZIP                       ZIP001
 1996/10/21 14:51:20  BACKUPINCR      4      1      1 ZIP                       ZIP002
 1996/10/25 13:12:05  BACKUPFULL      5      0      1 4MM                       LIBR03
```

You can see that a new backup series has been started with this full backup and has used volume LIBR03 in the library. You can also look at the contents of the tape library, using the QUERY LIBVOLUME command:

```
adsm> query libvolume
Library Name     Volume Name     Status               Last Use
------------     -----------     -----------------    ---------
4MMLIB           LIBR01          Private              Data
4MMLIB           LIBR02          Private              Data
4MMLIB           LIBR03          Private              DbBackup
4MMLIB           LIBR04          Scratch
```

You can see that LIBR03 is now in Private status and has been used for a database backup. Now use the DELETE VOLUMEHISTORY command to return this volume to scratch status. Use the date, time, and type parameters to limit the volumes that are deleted. Delete all database backups that have taken place up to the current date and time:

```
adsm> delete volhistory todate=today totime=now type=dbbackup
```

Attention!

This is an example to illustrate the process. We do not recommend that you delete all database backups. Always maintain one or two database backups.

Use the QUERY LIBVOLUME command to check that volume LIBR03 has returned to scratch status:

```
Library Name     Volume Name     Status               Last Use
------------     -----------     -----------------    ---------
4MMLIB           LIBR01          Private              Data
4MMLIB           LIBR02          Private              Data
4MMLIB           LIBR03          Scratch
4MMLIB           LIBR04          Scratch
```

It is important to periodically delete old database backup volumes with the DELETE VOLHISTORY command. Database backups, both full and incremental, always use a new volume. Over time a large number of volumes will be used for these database backups. These volumes cannot be reused for further database backups or for other server tasks such as in storage pools. Deleting old backups from the volume history frees the volumes which can then be reused. If a regular series of database backups is scheduled, the DELETE VOLHISTORY command should also be scheduled with appropriate TODATE= parameters. This approach will maintain recent database backups while returning old backup volumes to scratch status, allowing them to be reused.

Chapter 6. Storage Pool Availability

ADSM uses copy storage pools to make copies of data in primary storage pools. Copy storage pools are used to recover from a media or device failure affecting the primary storage pools. Offsite copy storage pools are used for server disaster recovery. In this chapter we provide examples of:

- Configuring copy storage pools
- Storage pool backup
- Storage pool recovery

6.1 Copy Storage Pool Configuration

Backing up and recovering storage pool data require that copy storage pools be defined. For our examples of storage pool backup and recovery, we created the storage pool configuration in Figure 52. The arrows represent the logical data flow for the various backup storage pool commands we will use.

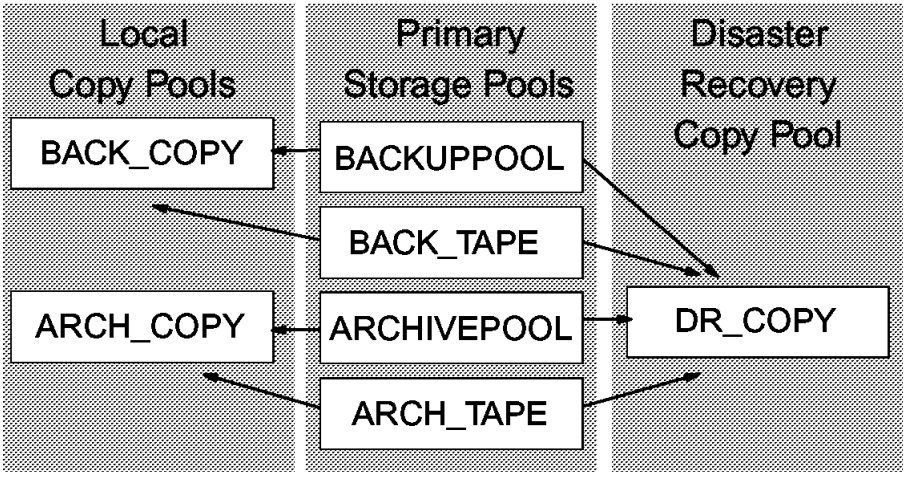


Figure 52. Copy Storage Pool Configuration

Our sample setup has four primary storage pools. We define two sets of copy storage pools, one for local and one for disaster recovery. Locally we have two copy storage pools, BACK_COPY and ARCH_COPY. BACK_COPY stores copies of backup data and ARCH_COPY stores copies of archive data.

Multiple copy storage pools are defined for disaster recovery purposes. If only one copy pool were defined and stored offsite, you could not perform

recovery for local media failure. However, if you retrieved the tapes from the disaster recovery vault and brought them onsite for media recovery purposes, you would have an exposure in the event of a server disaster.

Copy storage pools are defined in the same way as primary pools, using the DEFINE STGPOOL administrator command. On the basis of the configuration in Figure 52 on page 97, execute the following commands:

```
adsm> define stgpool ARCH_COPY 4mm pooltype=copy maxscr=100
ANR2200I Storage pool ARCH_COPY defined (device class 4MM).

adsm> define stgpool BACK_COPY 4mm pooltype=copy maxscr=100
ANR2200I Storage pool BACK_COPY defined (device class 4MM).

adsm> define stgpool DR_COPY 8mm pooltype=copy maxscr=100 reuse=35
ANR2200I Storage pool DR_COPY defined (device class 8MM).
```

This defines three copy storage pools. The ARCH_COPY and BACK_COPY pools are local pools defined with the 4MM device class. Volumes created in these pools will be in the Aiwa library.

The offsite DR_COPY pool is defined with the 8MM device class for use with the manual 8MM drive. The reuse delay (REUSE=35) parameter for the DR_COPY pool specifies that after all data on this tape has expired, 35 days must pass before the tape is returned to scratch status and can be reused. This is a safety precaution for these offsite tapes. If an old version of the database is restored during an offsite recovery, the expired files on tape might still be needed. Keeping them offsite for a further 35 days should ensure that they will not be returned to scratch status before they might be needed. When defining copy storage pools, you do not have to update the existing primary pools. They are stand-alone pools isolated from the primary pools.

To verify the copy storage pool definitions, issue this command:

```
adsm> query stgpool * pooltype=copy

Storage      Device       Estimated  %Util  %Migr  High   Low  Next
Pool Name    Class Name    Capacity                Mig%  Mig%  Storage
                               (MB)                            Pool
-----------  ----------  ----------  -----  -----  ----  ----  ---------
ARCH_COPY    4MM               0.0    0.0
BACK_COPY    4MM               0.0    0.0
DR_COPY      8MM               0.0    0.0
```

Having defined the copy storage pools, you can now back up the primary pools.

6.2 Storage Pool Backup

ADSM can maintain multiple copies of workstation files in storage pools for protection against media failure and for server disaster recovery. Back up the primary storage pools, using the BACKUP STGPOOL command:

```
adsm> BACKUP STGPOOL  storage_pool_name  copy_pool_name
```

An administrator issues the backup storage pool command specifying the primary pool to be backed up and the destination copy storage pool. Each primary storage pool must be backed up separately. However, multiple primary pools can be backed up to the same copy pool. The storage pool backup process is an incremental process; only new and changed files in a storage pool are copied to the copy storage pool. This capability allows for high availability with minimum server overhead.

Figure 52 on page 97 shows our sample storage pool configuration where we have two different sets of copy storage pools, one stored onsite to protect against media failure and one offsite for disaster recovery purposes. To back up the primary storage pools, you have to issue a series of backup storagepool commands:

```
adsm> backup stgpool BACKUPPOOL    BACK_COPY
adsm> backup stgpool BACK_TAPE     BACK_COPY
adsm> backup stgpool ARCHIVEPOOL   ARCH_COPY
adsm> backup stgpool ARCH_TAPE     ARCH_COPY
adsm> backup stgpool BACKUPPOOL    DR_COPY
adsm> backup stgpool BACK_TAPE     DR_COPY
adsm> backup stgpool ARCHIVEPOOL   DR_COPY
adsm> backup stgpool ARCH_TAPE     DR_COPY
```

Each backup storagepool command starts a background process. You can monitor these processes with the QUERY PROCESS command:

```
adsm> query process

 Process Process Description  Status
  Number
-------- -------------------- -------------------------------------------
       5 Backup Storage Pool  Primary Pool BACK_TAPE, Copy Pool BACK_COPY
                               Files Backed Up: 22, Bytes Backed Up: 2,11
                               Unreadable Files: 0, Unreadable Bytes: 0.
                               Current File (bytes): 304,676

                               Current input volume: LIBR01.

                               Current output volume: LIBR03.
```

The command output indicates that the BACK_TAPE to BACK_COPY backup process is running. It is backing up the contents of volume LIBR01 in the BACK_TAPE pool to volume LIBR03, which is a scratch volume that has been added to the BACK_COPY storage pool. This process can also be monitored at the server console or in the activity log:

```
 ...
ANR1210I Backup of primary storage pool BACKUPPOOL to copy storage pool
BACK_COPY started as process 4.
ANR8337I 4MM volume LIBR03 mounted in drive DRIVE0 (MT3).
ANR1212I Backup process 4 ended for storage pool BACKUPPOOL.
ANR1214I Backup of primary storage pool BACKUPPOOL to copy storage pool BACK_COPY
has ended.  Files Backed Up: 5, Bytes Backed Up: 102400, Unreadable Files: 0,
Unreadable Bytes: 0.
ANR1210I Backup of primary storage pool BACK_TAPE to copy storage pool BACK_COPY
started as process 5.
ANR1228I Removable volume LIBR01 is required for storage pool backup.
ANR8324I 4MM volume LIBR01 is expected to be mounted (R/W).
ANR8337I 4MM volume LIBR01 mounted in drive DRIVE1 (MT4).
ANR1212I Backup process 5 ended for storage pool BACK_TAPE.
ANR1214I Backup of primary storage pool BACK_TAPE to copy storage pool BACK_COPY
has ended.  Files Backed Up: 2642, Bytes Backed Up: 333075902, Unreadable Files:
0, Unreadable Bytes: 0.
 ...
```

To maintain these local and disaster recovery copy storage pools, run the BACKUP STGPOOL command daily after workstation backup operations. To schedule them use an administrative schedule.

Note

To back up tape primary pools, a minimum of two tape drives is required, one for the primary pool volume and one for the copy pool volume. With single drive configurations, it is not possible to back up primary tape pools to tape copy pools.

6.2.1 Moving Copy Pool Volumes Offsite

The process of backing up storage pools is done locally to the server, including copy pools for offsite recovery. When storage pool backups are run, the server adds new copy pool data to existing volumes in the destination copy pool. When restoring a primary pool, the server assumes that all the copy storage pool volumes are onsite and available. For our DR_COPY disaster recovery pool we want to move the volumes offsite. Once offsite, they will not be used for subsequent backups and can be stored in a vault for safe keeping.

To move these volumes offsite, you have to change their access mode to *offsite* and then physically move them offsite. Volumes with an access mode of offsite will not be reused by the server until the mode is set back to readwrite or readonly. The following administrator command changes the access mode of all DR_COPY copy storage pool volumes to offsite and their location information to vault, if the volumes are full or filling, and their access mode is readwrite or readonly:

```
adsm> update volume * acc=OFF loc='VAULT' wherestg=DR_COPY
     whereacc=READW,READO  wherest=FUL,FIL
```

The next time a storage pool backup is performed to the DR_COPY pool, a new scratch volume will be added to the DR_COPY pool. For our DR_COPY pool, you can schedule the above command to run periodically. Having updated the offsite copy pool volumes, you can determine which volumes to physically move offsite with the QUERY VOLUME command using the access= and stgpool= parmameters:

```
adsm> query volume access=offsite stgpool=DR_COPY

Volume Name              Storage      Device      Estimated  %Util   Volume
                         Pool Name    Class Name   Capacity          Status
                                                       (MB)
-----------------------  -----------  ----------  ---------  -----  -------
COPY01                   DR_COPY      8MM           4,944.0   55.4  Filling
COPY02                   DR_COPY      8MM           4,944.0   37.2  Filling
```

These volumes can now be physically moved to an offsite vault for future recovery of the server.

Library Volumes

Volumes updated to offsite that are located in SCSI libraries must be checked out of the library with the CHECKOUT LIBVOLUME command before you move them offsite.

6.2.2 Tape Reclamation

Over time the contents of storage pool volumes will become fragmented because files are deleted on workstations, older backup versions are expired, and files are moved. This reusable space is tracked in the database and can be reclaimed by consolidating the data onto other volumes within the storage pool. This process is controlled by specifying a reclamation percentage. When that percentage of free space is reached on a volume, the volume′s contents will be reclaimed onto other volumes in the same storage pool.

This process of space reclamation also works for offsite copy storage pools. When you define a copy storage pool, the reclamation threshold is set by default to 100%. This value causes reclamation never to occur. It is possible to perform reclamation of offsite copy storage pool volumes by altering the reclamation threshold periodically. For volumes marked as offsite this process does not require that the offsite volume being reclaimed is mounted. The server determines which files are still valid on that volume and copies them from the original primary pool to new volumes in the offsite pool that have not been marked as offsite. If all volumes are offsite, the process will use a new scratch volume.

One approach to managing this reclamation process is to schedule a command to lower the threshold once a week and shortly thereafter raise the threshold back to 100%. To lower the threshold issue the following command:

```
adsm> update stgpool DR_COPY reclaim=60
```

Reclamation will start immediately for full volumes with more than 60% of reclaimable data. Having estimated the time reclamation takes, define a second schedule that starts once a week after reclamation has finished to reset the threshold to 100%:

```
update stgpool DR_COPY reclaim=100
```

Offsite Reclamation

Offsite storage pools should always be defined with an appropriate REUSEDELAY value, preventing reclaimed volumes from being returned to scratch status immediately. This protects against the situation where a reclaimed volume might still be required when an old database backup is used for server recovery.

6.3 Storage Pool Recovery

Storage pool backups are used to protect against local hardware or media failures and to provide disaster recovery with offsite copy pools. The difference between local and offsite copy storage pools is the access mode of the volumes in the pool. Typically the access mode is set to read/write, in which case the server assumes that the volume can be mounted when required. For offsite copy pools, the access mode of its volumes must be updated to offsite. In this mode the server will not attempt to mount the volumes until they have been updated by the administrator to read/write mode. This process of updating volumes to go offsite is performed by the administrator.

With local copy pools, access to client workstation data is automatic if a primary pool volume fails. If the server cannot retrieve a client file from a primary storage pool, it automatically retrieves it from a copy pool, assuming that the copy pool volumes are available and in read/write mode.

To illustrate this we simulated the failure of a storage pool volume. We stopped the server and deleted a volume, DATA1.DSM, in the BACKUPPOOL disk primary storage pool. Restarting the server resulted in a message indicating that the deleted volume could not be varied online during server startup.

When a client workstation attempts to restore a file residing on the deleted volume, the following messages will appear on the server console or in the activity log:

```
ANR1421W Read access denied for volume C:\WIN32APP\IBM\ADSM\SERVER\DATA1.DSM -
volume offline.
ANR8337I 4MM volume LIBR03 mounted in drive DRIVE1 (MT4).
```

The first warning message is that the required volume, DATA1.DSM, is offline. The second message is that the server is mounting volume LIBR03. This is the BACK_COPY volume that contains the backup copy of the client workstation file on DATA1.DSM. You will see a message saying that the

restore is waiting for offline media. This is the server restoring the file from the copy storage pool. When the volume for the copy storage pool is mounted, the client restore will continue normally. This example is for a disk volume. The process is the same for a tape volume failure.

This dynamic switching to the copy storage pool ensures that client workstation data can always be restored in the event of a primary storage pool hardware or media failure. This ability to restore data from a copy pool only works for client workstations restoring or retrieving data. New workstation data being backed up to the server cannot be written directly to a copy storage pool. It must always be written initially to a primary pool and then backed up to the copy pool. In our example above, new backup data would be written to other volumes in the disk BACKUPPOOL if configured, or if not, to the next primary pool in the hierarchy, BACK_TAPE.

In either case you would want to recover the storage pool volume that has been lost. This can be done with the RESTORE VOLUME command.

6.3.1 Restoring Volumes

To recover storage pool volumes, use the RESTORE VOLUME command, which restores all of the files that resided on the lost volume from copies stored on copy storage pool volumes. The command restores files to the same storage pool as the lost volume, but not to the same volume. If there is sufficient capacity on the remaining volumes in the primary storage pool to hold the data from the lost volume, the data is restored to those volumes. If there is insufficient space or in the case where the lost volume was the only volume in the primary pool, the data is restored to the next primary pool in the hierarchy.

6.3.1.1 Restoring Disk Volumes

To restore the data that was on disk volume DATA1.DSM, we formatted another disk volume, DATA2.DSM:

```
dsmfmt -data C:\WIN32APP\IBM\ADSM\SERVER\DATA2.DSM 25
```

We then defined it to the BACKUPPOOL:

```
adsm> define volume BACKUPPOOL C:\WIN32APP\IBM\ADSM\SERVER\DATA2.DSM
```

Once we created additional space in our primary pool, we restored the contents of DATA1.DSM with the RESTORE VOLUME command. The copy

pool to be restored from can optionally be specified with the COPY parameter:

```
adsm> restore volume data1.dsm copy=BACK_COPY

ANR2041W This command attempts to restore all files in storage pool BACKUPPOOL
which reside on one of the volumes specified in the command; existing
references to files on these volumes will be deleted from the database after
the files have been restored.
Do you wish to proceed? (Yes/No) yes

ANR2114I RESTORE VOLUME: Access mode for volume
C:\WIN32APP\IBM\ADSM\SERVER\DATA1.DSM updated to "destroyed".
ANR2110I RESTORE VOLUME started as process 8.
```

The restore volume process updates the volume access mode to “destroyed” and then starts restoring the data. You can monitor the progress of the restore on the server console or in the activity log:

```
ANR1232I Restore of volumes in primary storage pool BACKUPPOOL started a s
process 8.
ANR1254I Removable volume LIBR03 required for restore processing.
ANR8324I 4MM volume LIBR03 is expected to be mounted (R/W).
ANR8337I 4MM volume LIBR03 mounted in drive DRIVE1 (MT4).
ANR1235I Restore process 8 ended for volumes in storage pool BACKUPPOOL.
ANR1240I Restore of volumes in primary storage pool BACKUPPOOL has ended . Files
Restored: 25, Bytes Restored: 5469894, Unreadable Files: 0, Unreadable Bytes:
0.
ANR2208I Volume C:\WIN32APP\IBM\ADSM\SERVER\DATA1.DSM deleted from storage pool
BACKUPPOOL.
```

On completion of the restore, the destroyed volume is automatically deleted from the storage pool.

Volume Status

When performing a restore volume, the disk volume being restored must be offline to the server but still defined in the storage pool. Failure to vary the volume offline will cause the restore to cancel. The volume must not be deleted before it is restored. Deleting the volume also deletes all references to its data in the copy storage pools, thereby making recovery impossible.

6.3.1.2 Restoring Tape Volumes

The process for restoring tape storage pool volumes is basically the same as for disk volumes. In this example we recover the contents of volume LIBR02 in BACK_TAPE pool, using the BACK_COPY pool:

```
adsm> restore volume libr02 copy=BACK_COPY

ANR2041W This command attempts to restore all files in storage pool BACK_TAPE
which reside on one of the volumes specified in the command; existing
references to files on these volumes will be deleted from the database after
the files have been restored.
Do you wish to proceed? (Yes/No) yes

ANR2114I RESTORE VOLUME: Access mode for volume LIBR02 updated to "destroyed"
ANR2110I RESTORE VOLUME started as process 11.
```

The restore volume process again updates the volume access mode to "destroyed" and then starts restoring the data:

```
ANR1232I Restore of volumes in primary storage pool BACK_TAPE started as
11.
ANR1254I Removable volume LIBR03 is required for restore processing.
ANR8337I 4MM volume LIBR03 mounted in drive DRIVE0 (MT3)
  ....
ANR1235I Restore process 11 ended for volumes in storage pool BACK_TAPE.
ANR1240I Restore of volumes in primary storage pool BACK_TAPE has ended. Files
Restored: 2725, Bytes Restored: 339864327, Unreadable Files: 0, Unreadable
Bytes: 0.
ANR2208I Volume LIBR02 deleted from storage pool BACK_TAPE.
ANR1341I Scratch volume LIBR02 has been deleted from storage pool BACK_TAPE
```

On completion of the restore, the destroyed volume is automatically deleted from the storage pool and returned to scratch status. We can confirm this by querying the library volumes:

```
adsm> query libvolume

Library Name     Volume Name     Status                Last Use
------------     -----------     -----------------     ---------
4MMLIB           LIBR01          Private               Data
4MMLIB           LIBR02          Scratch
4MMLIB           LIBR03          Private               Data
4MMLIB           LIBR04          Scratch
4MMLIB           LIBR05          Scratch
4MMLIB           LIBR06          Scratch
```

Volume LIBR02 has been returned to scratch status. This volume can now be checked out of the library to ensure that it is not reused.

Chapter 7. Server Recovery Example

In this chapter we describe how a server can be recovered by using database backups and copy storage pools.

To implement an effective disaster recovery plan for the server, you must carry out these tasks:

- Gather server configuration data required for recovery on a regular basis and maintain it in a safe location.
- Perform regular database and storage pool backups and take the appropriate volumes offsite to a secure location.
- Test the recovery process.

7.1 Server Preparation

The recovery of an ADSM server requires preparation and planning. In this section we look at gathering the appropriate server configuration information and automating server backups through the use of administrative schedules.

7.1.1 Server Configuration Data

To enable effective recovery of a server, you must have up-to-date information detailing the server configuration. This information includes:

- DSMSERV.OPT
- Latest volume history file
- Current device configuration file
- Database and recovery log configuration
- Database backup volumes
- DR_COPY copy storage pool volumes

All of this information is readily available while the server is running. You can obtain it by looking at files on the server or running administrator commands. Without a running server, this information is unavailable, and without the information, server recovery is impossible.

To automate the gathering of information, use an administrative macro containing the various commands that provide the required information:

```
query option
query status
query db format=detail
query dbvol format=detail
query log format=detail
query logvol format=detail
query stgpool format=detail
query volhhist type=dbb
query vol format=detail
query dbb format=detail
```

You can save these commands in a macro file and run the macro by starting an administrative client session in batch mode. In our example we created a macro called OFFSITE.MAC and ran it from a Windows NT command prompt:

```
dsmadmc -id=admin -pass=passw macro OFFSITE.MAC  > OUT_FILE_NAME
```

This command redirects the macro output to a file called OUT_FILE_NAME. We suggest that you run such a macro on a regular basis and save the output report in a secure place. For an example of the output of this macro, see A.3, “Sample SERVER.REP” on page 131.

Note

The macro must be run in batch mode for the output to be formatted correctly. Running a macro and redirecting the output to file from an interactive administrator session results in garbled output.

The above administrator macro provides the basic information required. However, for our example we took this further and created a batch file, OFFSITE.CMD, that:

1. Copies DSMSERV.OPT
2. Runs the OFFSITE.MAC macro, directing the output to a file called SERVER.REP
3. Backs up the volume history file
4. Backs up the device configuration file

The OFFSITE.CMD batch file can be tailored with configuration-specific information such as the server directory, client directories, and the drive and directory information where the output files will be stored. Appendix A, “Sample Recovery Command File and Report” on page 129 contains an example of this batch file.

This batch file can be run manually or scheduled with the ADSM scheduler. As an operating system command, it must be scheduled as a client schedule rather than as an administrative schedule.

7.1.2 Server Scheduling

Some effort in designing an appropriate schedule is required to enable the server to complete all schedules within a defined timeframe. We used the following example to show how a combination of client backups and server tasks can be scheduled. The objective of the schedule is to run nightly client workstation backups, followed by local and disaster recovery storage pool backups, followed by database backups. Figure 53 shows the relative timings of our client and server scheduled operations.

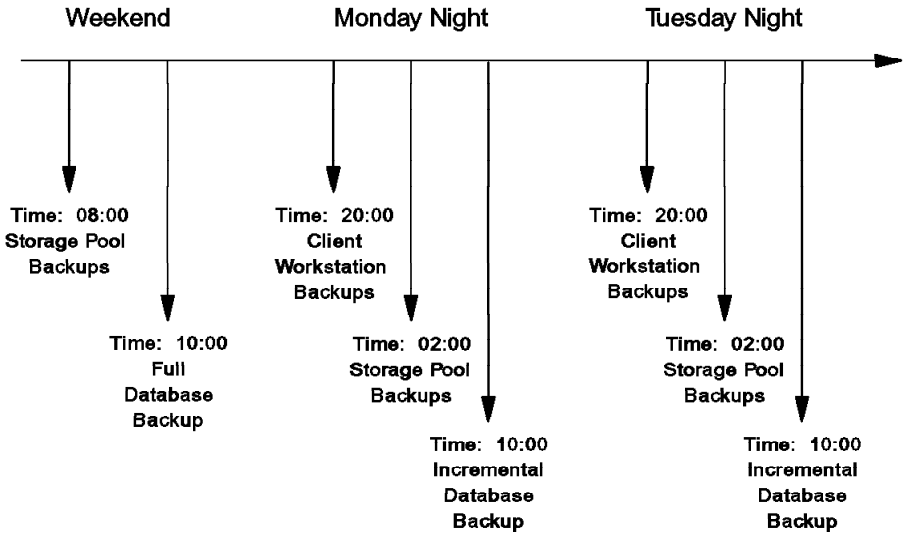


Figure 53. Client and Server Scheduled Workload

The scheduled operations commence with storage pool and full database backups on the weekend. The subsequent weekday scheduled operations start with client workstation backups performed every evening at 8:00 p.m. These backups run every day; we assumed that they complete by 1:00 a.m. the following day. The next step is to back up the primary storage pools to the local and disaster recovery copy storage pools.

To perform these storage pool backups you can schedule individual administrative schedules that back up the storage pools:

```
adsm> define schedule backuppool type=admin cmd=backup stg backuppool back_copy
 active=yes description="Storage pool backup: backuppool to back_copy"
 starttime=02:00 dayofweek=weekday
```

This schedule will back up the BACKUPPOOL disk pool to the BACK_COPY copy storage pool every weekday morning a 2:00 a.m. You must repeat this method of scheduling for every primary pool to be backed up.

An alternative is to create an administrative macro that performs all storage pool backups:

```
backup stgpool backuppool   back_copy
backup stgpool back_tape    back_copy
backup stgpool archivepool  arch_copy
backup stgpool arch_tape    arch_copy
backup stgpool backuppool   dr_copy
backup stgpool back_tape    dr_copy
backup stgpool archivepool  dr_copy
backup stgpool arch_tape    dr_copy
```

However, you cannot schedule this macro with an administrative schedule, which can only schedule individual commands. You schedule this type of macro by creating a command file that invokes the administrative client in batch mode to run the above macro:

```
dsmadmc -id="admin id" -password="secret" macro c:\stgcopy.mac
```

In this example the STGCOPY.CMD command file runs the above STGCOPY.MAC macro. To schedule this command, use the backup/archive scheduler running on the server by scheduling the following command:

```
adsm> define schedule standard daily_stgcopy type=client action=command
objects=c:\stgcopy.cmd startdate=10/15/1996 starttime=02:00:00
dayofweek=weekday
```

Finally we want to schedule the database backups. We want to ensure that all storage pool backups, to both the local and disaster recovery pools, complete before the database is backed up. Therefore we schedule the database backups for 8:00 a.m. We are going to run a full database backup during the weekend and incremental backups during the week. To do this we have to define three administrative schedules:

1. The first schedule defines a full database backup that runs at 10:00 a.m. on Sundays:

```
adsm> define schedule dbback_full type=admin cmd="BACKUP DB DEVC=ZIP TYPE=FULL"
      active=yes starttime=10:00:00 dayofweek=Sunday
```

2. The second schedule defines incremental database backups that will run at 10:00 a.m. on weekdays:

```
adsm> define schedule dbback_incr1 type=admin cmd="BACKUP DB DEVC=ZIP TYPE=INCR"
      active=yes starttime=10:00:00 dayofweek=Weekday
```

3. The third schedule defines a final incremental database backup at 10:00 a.m. on the following Saturday. This falls outside the weekday definition and must be scheduled separately:

```
adsm> define schedule dbback_incr2 type=admin cmd="BACKUP DB DEVC=ZIP TYPE=INCR"
      active=yes starttime=10:00:00 dayofweek=Saturday
```

These schedules produce a weekly series of database backups. Each series will consist of a full backup on Sunday, followed by six incremental backups during the week. Every Sunday a new backup series is started. Each of these database backups uses a new volume that cannot be used for other purposes. Because we are creating a weekly series, any database backups older than the latest full backup should no longer be required. To release the volumes used by these old backup series, we can schedule the DELETE VOLUMEHISTORY command to run daily after the database backups have completed:

```
adsm> define schedule dbback_del type=admin cmd="DELETE VOLHISTORY TODATE=-8
      TYPE=DBBackup" active=yes starttime=10:00:00 dayofweek=Any
```

This schedule will daily delete any database backup volumes from the volume history that are more than eight days old. By maintaining the last eight daily database backup volumes, we can be sure that we always have the last full backup and subsequent incrementals.

7.1.3 Moving Volumes Offsite

Once the daily schedules for storage pool and database backups have been defined, the volumes destined for offsite disaster recovery purposes must be moved offsite on a regular basis, preferably daily for maximum availability.

First we will move the volumes being created in the DR_COPY copy storage pool. Before they can be physically moved, their access mode must be updated to offsite:

```
adsm> update volume * acc=OFF loc='VAULT' wherestg=DR_COPY whereacc=READW
       wherest=FUL,FIL
```

The UPDATE VOLUME command updates any volume in the DR_COPY that has an access mode set to READWRITE and a status of FULL or FILLING. Any volumes matching these criteria will have their access mode set to OFFSITE and their location information updated to "VAULT." This prevents the server from requesting that these volumes be mounted the next time the primary storage pools are backed up to the DR_COPY pool, allowing them to be physically moved offsite. Running this command results in message ANR2207I being issued for each volume updated:

```
ANR2207I Volume COPY01 updated.
ANR2207I Volume COPY02 updated.
```

Once these volumes are updated, the next storage pool backup will use a new scratch volume.

The UPDATE VOLUME command can be scheduled to run following the storage pool backups, and the activity log can be monitored to determine which volumes are to be moved offsite.

The second set of volumes that should be moved offsite is the database backup series. Volumes used for database backups are not tracked in the same manner as those used in storage pools. The QUERY VOLUME and UPDATE VOLUME commands apply only to volumes used in storage pools. For database backups the volume history contents must be monitored by viewing the volume history backup file or using the QUERY VOLHISTORY command.

```
adsm> query volhistory type=dbbackup

       Date/Time: 10/08/1996 16:25:56
     Volume Type: BACKUPFULL
   Backup Series: 35
Backup Operation: 0
      Volume Seq: 1
    Device Class: ZIP
     Volume Name: ZIP001
 Volume Location:
```

In 7.1.2, "Server Scheduling" on page 109, to maintain a weekly database backup series we scheduled the DELETE VOLHISTORY command to delete any database backup volumes older than eight days. The above query

command therefore shows all volumes currently used for database backups that should be moved offsite. It is not necessary to update these volumes as the database backup process does not use existing volumes.

Over time the collection of offsite copy storage pool and database backup volumes will grow and expire. Data in the offsite copy storage pools will expire and the volumes will empty. This process of reclaiming space on offsite volumes is a logical one. The volumes do not have to be read; their contents are tracked and maintained in the server database. Once these volumes are empty, they can be returned onsite and reused as new scratch volumes. Periodically checks should be made for newly emptied volumes offsite:

```
adsm> query volume stgpool=dr_copy acc=offsite status=empty
```

This command will list any volumes in the DR_COPY pool that are offsite and empty. These can be brought back onsite and reused.

REUSEDELAY

If the REUSEDELAY parameter is used for the offsite storage pool, whatever value in days is specified must elapse after tapes become empty before they can be reused by the server. This is protection for circumstances where database backups are not performed in synchronization with storage pool backups. In this scenario if workstation files are expired from copy pool volumes, the changes will not be reflected in the database backups until the next database backup. If a server failure occurs between files being expired and the next database backup, workstation recovery could be jeopardized if the offsite volumes have already been returned onsite and reused. The REUSEDELAY parameter allows you to maintain volumes offsite for a specified number of days to prevent this from happening.

Database backup volumes are handled differently. The DELETE VOLHISTORY command allows deleted database backup volumes to be immediately reused. These volumes will accumulate offsite and should be brought back onsite where they can be used for subsequent database backups.

7.2 Server Recovery

In the previous chapters we have built up a set of concepts and processes that have taken us to make backups of all the ADSM server components. We

have discussed how to recover from the failure of single components, such as database or storage pool volume loss. In this section we present an example of recovery of the ADSM server in the event of a disaster. We must have previously stored database and storage pool backups offsite together with the server configuration information discussed in 7.1.1, “Server Configuration Data” on page 107. We also assume that the recovery machine has Windows NT installed and operational, the ADSM server and administrative client have been installed, and we have devices that can read the database backup and copy storage pool removable media.

We discuss the steps necessary in recovering the ADSM server at an offsite location by working through an example. In this example we assume you have access to all media and diskettes previously stored offsite.

7.2.1 Server Preparation

1. Locate the most recent server configuration data at the offsite recovery site. Table 1 lists the information required for the recovery.

Table 1. Server Recovery Configuration Files

File Name	Contents
DSMSERV.OPT	Server options file
SERVER.REP	Server configuration report. Contains detailed server configuration information for our original server: • Database size and volumes • Recovery log size and volumes • Server options and status • Storage pool configuration • Server volume inventory
VOLHIST.OUT	Volume history file
DEVCNFG.OUT	Backup device configuration file

2. Browse the volume history file:

```
*******************************************************************************
*
*               IBM AdStar Distributed Storage Manager Sequential Volume Usage History
*                                    Updated 12/04/1996 16:59:27
*
*     Operation          Volume   Backup Backup Volume Device                     Volume
*     Date/Time           Type    Series Oper.   Seq   Class Name                Name
*******************************************************************************
 1996/11/26 11:25:45  STGNEW          0      0      0 4MM                       LIBR01
 1996/11/26 11:38:50  STGNEW          0      0      0 4MM                       LIBR02
 1996/10/20 08:38:15  BACKUPFULL      4      0      1 ZIP                       ZIP001
 1996/11/26 13:21:38  STGNEW          0      0      0 4MM                       LIBR03
 1996/10/21 14:51:21  BACKUPINCR      4      1      1 ZIP                       ZIP002
 1996/11/27 14:25:27  STGNEW          0      0      0 8MM                       COPY01
 1996/12/04 14:19:01  STGNEW          0      0      0 8MM                       COPY02
 1996/10/22 16:27:50  BACKUPINCR      4      2      1 ZIP                       ZIP003
 1996/10/23 16:54:10  BACKUPINCR      4      3      1 ZIP                       ZIP004
```

Determine the latest database backup volumes. These volumes have volume types of BACKUPFULL and BACKUPINCR. The latest backup series will have the highest Backup Series number. These volumes will be needed for the recovery of the database. In our case the database backup volume names are ZIP001, ZIP002, ZIP003, and ZIP004.

3. Copy the original volume history, device configuration, and server options files to the new ADSM server directory, in our case, C:\WIN32APP\IBM\ADSM\server:

```
copy a:VOLHIST.OUT C:\WIN32APP\IBM\ADSM\SERVER
copy a:DEVCNFG.OUT C:\WIN32APP\IBM\ADSM\SERVER
copy a:DSMSERV.OPT C:\WIN32APP\IBM\ADSM\SERVER
```

4. Edit DSMSERV.OPT to ensure that the VOLUMEHistory and DEVCONFig entries match the names of the files copied from our recovery diskette in step 3:

```
 ...
*   VOLUMEHISTORY
VOLUMEHistory volhist.out
*
*   DEVCONFIG
DEVCONFig devcnfg.out
 ...
```

If the recovery server has different network addresses, you may have to check the networking parameters in DSMSERV.OPT. These parameters are not needed for the offsite ADSM server recovery. However, they will be needed when ADSM client workstations reconnect to the server.

5. Determine the size of the ADSM database and recovery log from the server report file, SERVER.REP, on the recovery diskette. Locate the output of the *q dbv f=d* and *q logv f=d* commands.

```
ANS5101I Server command: 'q dbv f=d'

Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\DB1.DSM
         Copy Status: Sync'd
Volume Name (Copy 2): D:\DBMIRROR\DB1.DSM
         Copy Status: Sync'd
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 32
Allocated Space (MB): 32
     Free Space (MB): 0

 ... Lines Omitted

ANS5101I Server command: 'q logv f=d'

Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\LOG1.DSM
         Copy Status: Sync'd
Volume Name (Copy 2):
         Copy Status: Undefined
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 48
Allocated Space (MB): 48
     Free Space (MB): 0

 ... Lines Omitted
```

From the above report, determine the database and recovery log volume names and sizes: 32 MB for the database and 48 MB for the recovery log. Before the database can be restored, the recovery ADSM server must be installed and operational with a database and log of these sizes. The ADSM installation creates by default the following database and recovery log files:

- DB1.DSM - 4 MB
- LOG1.DSM - 8 MB

These can be used with additional volumes formatted and added. Alternatively, the server could be reinstalled with new volumes of the required size, which we do in step 6.

6. If the recovery server has previously been installed, delete the database (DB1.DSM), log (LOG1.DSM), and DSMSERV.DSK files before reinstalling the server. Having done this the DSMSERV INSTALL command can be used to install and prepare the database and recovery log files for use by the server and database restore utility. You can create as many database volumes as you choose, provided the sum of the new volume sizes matches the sum of the original sizes. You can choose whatever file names you like for the database and recovery log; they do not have to be the same as the names used in the original ADSM server. In our

example we installed the server with one recovery log and database volume in the ADSM server directory:

```
C:\WIN32APP\IBM\ADSM\server>dsmserv install 1 log1.dsm 49 1 db1.dsm 33
ANR0900I Processing options file C:\WIN32APP\IBM\ADSM\server\dsmserv.opt.

ADSTAR Distributed Storage Manager for Windows NT
Version 2, Release 1, Level 0.12/0.12

Licensed Materials - Property of IBM

5639-A09 (C) Copyright IBM Corporation 1990, 1996. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 22:06:54 on Nov 18 1996.
Allocated space for C:\WIN32APP\IBM\ADSM\server\log1.dsm: 51380224 bytes.
Allocated space for C:\WIN32APP\IBM\ADSM\server\db1.dsm: 34603008 bytes.
ANR0300I Recovery log format started; assigned capacity 48 megabytes.
ANR0301I Recovery log format in progress; 4 megabytes of 48.
ANR0301I Recovery log format in progress; 8 megabytes of 48.
 ... lines omitted
ANR0301I Recovery log format in progress; 48 megabytes of 48.
ANR0302I Recovery log formatting took 50000 milliseconds.
ANR0303I   Format rate:         245.8  pages/second.
ANR0304I   Page service time:     4.1  ms.
ANR0305I Recovery log format complete.
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR0992I ADSM server installation complete.
```

This reinstalls the server, formatting the database and recovery log volumes.

7. You can now start the ADSM server to verify that the volumes are installed correctly. Start the ADSM server from an NT command prompt by entering DSMSERV, and issue query commands to display the database and recovery log volumes:

```
adsm> query dbvolume f=d
adsm> query logvolume f=d
```

Verify that the volume sizes displayed match those in the server report file, SERVER.REP.

You now have the recovery server installed with a database and recovery log of the required size. The original DSMSERV.OPT, VOLHIST.OUT, and

DEVCNFG.OUT files have been copied to the server directory. With these steps complete you can proceed with recovering the database.

7.2.2 Database Recovery

You are now ready to restore the database. Because you are restoring it on a new server you do not have the contents of the original server recovery log. Therefore you cannot perform a roll forward recovery; you have to do a point-in-time recovery with the latest available database backup series. In the example of a point-in-time recovery in 5.2.3, “Database Recovery” on page 85 we performed the recovery in two stages, restoring the individual volumes and then committing the restored database. In this example, we use another method: specifying the TODATE= parameter to perform the restore in one stage. The TODATE= parameter checks the volume history backup file for the latest database backup series before the specified date and prompts for all these volumes one after the other.

1. Issue the DSMSERV RESTORE DB command with the preview=yes option to confirm which volumes will be required for the restore:

```
C:\WIN32APP\IBM\ADSM\server>dsmserv restore db todate=today preview=yes
ANR0900I Processing options file C:\WIN32APP\IBM\ADSM\server\dsmserv.opt.

ADSTAR Distributed Storage Manager for Windows NT
Version 2, Release 1, Level 0.12/0.12

Licensed Materials - Property of IBM

5639-A09 (C) Copyright IBM Corporation 1990, 1996. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 22:06:54 on Nov 18 1996.
ANR0200I Recovery log assigned capacity is 48 megabytes.
ANR0201I Database assigned capacity is 32 megabytes.
ANR4600I Processing volume history file volhist.out.
ANR4620I Database backup series 1 operation 0 device class ZIP.
ANR4622I   Volume 1: ZIP001.
ANR4620I Database backup series 1 operation 1 device class ZIP.
ANR4622I   Volume 1: ZIP002.
ANR4620I Database backup series 1 operation 2 device class ZIP.
ANR4622I   Volume 1: ZIP003.
ANR4620I Database backup series 1 operation 3 device class ZIP.
ANR4622I   Volume 1: ZIP004.
```

2. Having confirmed and obtained the required volumes, you can run the DSMSERV RESTORE DB command without the preview option to recover the database (Figure 54 on page 119).

```
C:\WIN32APP\IBM\ADSM\server>dsmserv restore db todate=today
ANR0900I Processing options file C:\WIN32APP\IBM\ADSM\server\dsmserv.opt.

ADSTAR Distributed Storage Manager for Windows NT
Version 2, Release 1, Level 0.12/0.12

Licensed Materials - Property of IBM

5639-A09 (C) Copyright IBM Corporation 1990, 1996. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7800I DSMSERV generated at 22:06:54 on Nov 18 1996.
ANR0200I Recovery log assigned capacity is 48 megabytes.
ANR0201I Database assigned capacity is 32 megabytes.
ANR4600I Processing volume history file volhist.out.
ANR8324I REMOVABLEFILE volume ZIP001 is expected to be mounted (R/O).
ANR8324I REMOVABLEFILE volume ZIP002 is expected to be mounted (R/O).
ANR8324I REMOVABLEFILE volume ZIP003 is expected to be mounted (R/O).
ANR8324I REMOVABLEFILE volume ZIP004 is expected to be mounted (R/O).
ANR4620I Database backup series 1 operation 0 device class ZIP.
ANR4622I   Volume 1: ZIP001.
ANR4620I Database backup series 1 operation 1 device class ZIP.
ANR4622I   Volume 1: ZIP002.
ANR4620I Database backup series 1 operation 2 device class ZIP.
ANR4622I   Volume 1: ZIP003.
ANR4620I Database backup series 1 operation 3 device class ZIP.
ANR4622I   Volume 1: ZIP004.
ANR4634I Starting point-in-time database restore to date 12/04/1996 23:59:59.
ANR8326I 001: Mount REMOVABLEFILE volume ZIP001 R/O in drive ZIPDRIVE (F:) of
library ZIPLIB within 60 minutes.
ANR0300I Recovery log format started; assigned capacity 48 megabytes.
ANR0301I Recovery log format in progress; 4 megabytes of 48.
ANR0301I Recovery log format in progress; 8 megabytes of 48.
ANR0301I Recovery log format in progress; 12 megabytes of 48.
ANR0301I Recovery log format in progress; 16 megabytes of 48.
ANR0301I Recovery log format in progress; 20 megabytes of 48.
ANR0301I Recovery log format in progress; 24 megabytes of 48.
ANR0301I Recovery log format in progress; 28 megabytes of 48.
ANR0301I Recovery log format in progress; 32 megabytes of 48.
ANR8335I 001: Verifying label of REMOVABLEFILE volume ZIP001 in drive ZIPDRIVE
(F:).
ANR8328I 001: REMOVABLEFILE volume ZIP001 mounted in drive ZIPDRIVE (F:).
ANR8340I REMOVABLEFILE volume ZIP001 mounted.
ANR1363I Input volume ZIP001 opened (sequence number 1).
ANR4646I Database capacity required for restore is 32 megabytes.
ANR4638I Restore of backup series 1 operation 0 in progress.
ANR0301I Recovery log format in progress; 36 megabytes of 48.
ANR4639I Restored 496 of 556 database pages.
ANR0301I Recovery log format in progress; 40 megabytes of 48.
ANR1364I Input volume ZIP001 closed.
ANR4640I Restored 556 pages from backup series 1 operation 0.
ANR8317I The volume in REMOVABLEFILE drive ZIPDRIVE (F:) must be manually
ejected.
ANR0301I Recovery log format in progress; 44 megabytes of 48.
ANR0301I Recovery log format in progress; 48 megabytes of 48.
ANR0302I Recovery log formatting took 53000 milliseconds.
ANR0303I   Format rate:         231.8  pages/second.
ANR0304I   Page service time:     4.3  ms.
ANR0305I Recovery log format complete.
ANR8468I REMOVABLEFILE volume ZIP001 dismounted from drive ZIPDRIVE (F:) in
library ZIPLIB.
ANR8326I 002: Mount REMOVABLEFILE volume ZIP002 R/O in drive ZIPDRIVE (F:) of
library ZIPLIB within 60 minutes.
ANR8335I 002: Verifying label of REMOVABLEFILE volume ZIP002 in drive ZIPDRIVE
(F:).
ANR8328I 002: REMOVABLEFILE volume ZIP002 mounted in drive ZIPDRIVE (F:).
ANR8340I REMOVABLEFILE volume ZIP002 mounted.
ANR1363I Input volume ZIP002 opened (sequence number 1).
ANR4638I Restore of backup series 1 operation 1 in progress.
ANR1364I Input volume ZIP002 closed.
```

Figure 54 (Part 1 of 2). Database Point-in-Time Recovery

```
ANR4640I Restored 109 pages from backup series 1 operation 1.
ANR8317I The volume in REMOVABLEFILE drive ZIPDRIVE (F:) must be manually
ejected.
ANR8468I REMOVABLEFILE volume ZIP002 dismounted from drive ZIPDRIVE (F:) in
library ZIPLIB.
ANR8326I 003: Mount REMOVABLEFILE volume ZIP003 R/O in drive ZIPDRIVE (F:) of
library ZIPLIB within 60 minutes.
ANR8335I 003: Verifying label of REMOVABLEFILE volume ZIP003 in drive ZIPDRIVE
(F:).
ANR8328I 003: REMOVABLEFILE volume ZIP003 mounted in drive ZIPDRIVE (F:).
ANR8340I REMOVABLEFILE volume ZIP003 mounted.
ANR1363I Input volume ZIP003 opened (sequence number 1).
ANR4638I Restore of backup series 1 operation 2 in progress.
ANR4639I Restored 496 of 707 database pages.
ANR1364I Input volume ZIP003 closed.
ANR4640I Restored 707 pages from backup series 1 operation 2.
ANR8317I The volume in REMOVABLEFILE drive ZIPDRIVE (F:) must be manually
ejected.
ANR8468I REMOVABLEFILE volume ZIP003 dismounted from drive ZIPDRIVE (F:) in
library ZIPLIB.
ANR8326I 004: Mount REMOVABLEFILE volume ZIP004 R/O in drive ZIPDRIVE (F:) of
library ZIPLIB within 60 minutes.
ANR8335I 004: Verifying label of REMOVABLEFILE volume ZIP004 in drive ZIPDRIVE
(F:).
ANR8328I 004: REMOVABLEFILE volume ZIP004 mounted in drive ZIPDRIVE (F:).
ANR8340I REMOVABLEFILE volume ZIP004 mounted.
ANR1363I Input volume ZIP004 opened (sequence number 1).
ANR4638I Restore of backup series 1 operation 3 in progress.
ANR4640I Restored 113 pages from backup series 1 operation 3.
ANR0306I Recovery log volume mount in progress.
ANR4641I Sequential media log redo pass in progress.
ANR4642I Sequential media log undo pass in progress.
ANR1364I Input volume ZIP004 closed.
ANR8317I The volume in REMOVABLEFILE drive ZIPDRIVE (F:) must be manually
ejected.
ANR4644I A full backup will be required for the next database backup operation.
ANR4635I Point-in-time database restore complete, restore date 12/04/1996
16:54:10.
ANR4645I The restore date reflects the most recent backup available up to the
specified TODATE.
ANR8468I REMOVABLEFILE volume ZIP004 dismounted from drive ZIPDRIVE (F:) in
library ZIPLIB.
```

Figure 54 (Part 2 of 2). Database Point-in-Time Recovery

On completion of the restore, messages indicate the date and time to which the database was restored.

3. You can now restart the server and verify that the server database has been restored correctly. Issue commands such as QUERY NODE or QUERY VOLUME to verify that the server configuration is the same as the lost server.
4. If the device configuration in the disaster recovery site is not the same as that in the original site, new libraries and drives will have to be defined and the old device classes updated to point to the new libraries.

Having successfully recovered the database, you next have to recover the storage pools.

7.2.3 Storage Pool Recovery

With the new server operational, you can recover the storage pools in two stages:

1. Update previous storage pool volumes
2. Restore primary storage pools

7.2.3.1 Update Volumes

The restored database contains information about the storage pools and their volumes. The original primary pool volumes are no longer available and the server must be updated to reflect this. In addition, the DR_COPY pool volumes, which were previously offsite, are now available for use by client workstations to restore files and for recovering the primary storage pools. Again the server must be updated to reflect this:

1. No immediate action is required for disk storage pool volumes. When the server starts, it detects that the volumes are not available and marks them as offline.
2. You must tell the server which local storage pool volumes have been lost and are no longer available. Use the UPDATE VOLUME command to change the volume access mode to unavailable for all volumes in all storage pools that were lost:

```
adsm> update volume * acc=unavailabe wheredevc=4mm
ANR2207I Volume LIBR01 updated.
ANR2207I Volume LIBR02 updated.
ANR2207I Volume LIBR03 updated.
```

   This command updates any volume that used the 4MM device class with an access mode of unavailable. This includes all storage pool volumes in our Aiwa library and primary tape and local copy storage pool volumes.

3. The above command updates all the volumes that were used in the primary and local copy storage pools as unavailable. In addition to these, there are scratch volumes still defined in the Aiwa library that are no longer available. These must be removed to prevent the server from attempting to mount a scratch volume that no longer exists. To determine which scratch volumes are still defined in the library, use the QUERY LIBVOLUME command:

```
adsm> query libvolume 4MMLIB

Library Name     Volume Name     Status                Last Use
------------     -----------     ------------------    ---------
4MMLIB           LIBR01          Private               Data
4MMLIB           LIBR02          Private               Data
4MMLIB           LIBR03          Private               Data
4MMLIB           LIBR04          Scratch
4MMLIB           LIBR05          Scratch
4MMLIB           LIBR06          Scratch
```

The command output shows three volumes, LIBR01, LIBR02, and LIBR03, used as storage pool volumes. These were updated with a mode of unavailable in step 2. It also shows three scratch volumes: LIBR04, LIBR05, and LIBR06. These volumes are no longer available, and the server must be updated to reflect this. Use the CHECKOUT LIBVOLUME command:

```
adsm> checkout libvolume 4mmlib libr04 checklabel=no force=yes
adsm> checkout libvolume 4mmlib libr05 checklabel=no force=yes
adsm> checkout libvolume 4mmlib libr06 checklabel=no force=yes
```

The checklabel=no parameter is used to stop the server from mounting the volume to read the label. Even with this parameter the server will still attempt to read the barcode label if this function is supported by the library. The force=yes parameter allows the server to check out the volume even if an attempt is made to read the label or barcode and an error occurs.

All volumes defined in the primary and local copy pools have been marked as unavailable and any remaining scratch volumes defined in the automated library have also been removed. The server will not attempt to mount the volumes that are no longer available. When the ADSM server tries to access a client node file that is stored on a volume that is marked as unavailable, it retrieves a copy of the file from a copy storage pool, if available. In our case the local copy storage pool volumes are now unavailable, so the ADSM server must retrieve the files from the DR_COPY copy storage pool volumes. For this to work the volumes in this pool must be made available to the server.

4. This step determines the status of the DR_COPY pool volumes and updates their access mode to readwrite as appropriate. We can check which DR_COPY volumes are defined to the server with the QUERY VOLUME command:

```
adsm> query volume stgpool=dr_copy
```

This command will give us a list of all volumes used in the DR_COPY pool. Our next assumption is that there may have been volumes in the DR_COPY storage pool that were still onsite at the time of the disaster. These will have been lost with the rest of the server volumes. We now have to match the actual offsite DR_COPY volumes with the volumes defined to the server. The above QUERY VOLUME command will list all volumes defined in the DR_COPY pool. You can reissue the above command to determine which of these volumes have an access mode of readwrite:

```
adsm> query volume access=readw stgpool=dr_copy
```

Any volumes that have the access field set to readwrite we assume were still onsite and in use. These will also have been lost with the other storage pool volumes and must be deleted. Use the DELETE VOLUME command with the discarddata=yes parameter:

```
adsm> delete volume "VOLUME" discarddata=yes
```

Once the lost volumes have been deleted, the only volumes remaining in the DR_COPY pool are those offsite.

5. You must now tell the server that these offsite volumes are now onsite and can be accessed:

```
adsm> update volume * acc=readwrite wherestg=dr_copy
```

This command updates all the offsite volumes to readwrite, making them mountable by the server.

Now client workstations can connect to the server and recover their data if required from the DR_COPY pool. They will not be able to back up or archive any new data to the server as we have no primary storage pool volumes defined. Before we can restart backing up the workstation clients, we must recover the primary storage pools.

7.2.3.2 Restore Primary Storage Pools

In 6.3.1, "Restoring Volumes" on page 104 we show examples of restoring disk and tape volumes with the RESTORE VOLUME command. That command restores individual volumes that have been lost. In the examples that follow we use the RESTORE STGPOOL command, which restores all volumes in a storage pool. First we restore our primary disk pools, then our primary tape pools:

1. The RESTORE STGPOOL command restores from a copy storage pool the contents of any volume in the primary pool that has an access mode

of destroyed. The data is restored to other volumes in the storage pool. First we have to format volumes with different names for our disk pools. At a command prompt use the DSMFMT command:

```
dsmfmt back01.dsm 25
dsmfmt arch01.dsm 25
```

Once the new volumes are formatted, they have to be defined in the disk storage pools to be recovered:

```
adsm> define volume backuppool back01.dsm
adsm> define volume archivepool arch01.dsm
```

Once new disk volumes have been created and defined, the old volumes that have been lost must be updated with an access mode of destroyed. When the server was started after the database recovery, it attempted to vary online all of the disk volumes defined, but it could not because the volumes did not exist. As a consequence, the server set their access mode to offline. These must now be updated to destroyed. To determine which volumes to update, use the QUERY VOLUME command:

```
adsm> query volume  status=offline

Volume Name               Storage      Device      Estimated  %Util   Volume
                          Pool Name    Class Name  Capacity           Status
                                                        (MB)
------------------------  -----------  ----------  ---------  -----  --------
C:\WIN32APP\IBM\ADSM\SER  BACKUPPOOL   DISK              0.0    0.0  Off-Line
 VER\DATA01.DSM
C:\WIN32APP\IBM\ADSM\SER  ARCHIVEPOOL  DISK              0.0    0.0  Off-Line
 VER\DATA02.DSM
```

To update the offline disk volumes with an access mode of destroyed, use the UPDATE VOLUME command:

```
adsm> update volume * acc=destroyed wherestatus=offline
ANR2207I Volume C:\WIN32APP\IBM\ADSM\SERVER\DATA01.DSM updated.
ANR2207I Volume C:\WIN32APP\IBM\ADSM\SERVER\DATA02.DSM updated.
```

Having updated the volumes, you can proceed with the restore, using the RESTORE STGPOOL command:

```
adsm> restore stgpool backuppool

ANR1230I Restore of primary storage pool BACKUPPOOL started as process 13

ANR1254I Removable volume COPY02 is required for restore processing.
ANR8324I 8MM volume COPY02 is expected to be mounted (R/W).
ANR8326I 007: Mount 8MM volume COPY02 R/W in drive DRIVE1 (MT1.0.0.1) of library
8MMLIB within 60 minutes.
ANR8335I 007: Verifying label of 8MM volume COPY02 in drive DRIVE1 (MT1.0.0.1).
ANR8328I 007: 8MM volume COPY02 mounted in drive DRIVE1 (MT1.0.0.1).
ANR1234I Restore process 13 ended for storage pool BACKUPPOOL.
ANR1238I Restore of primary storage pool BACKUPPOOL has ended.  Files Restored:
22, Bytes Restored: 2117190, Unreadable Files: 0, Unreadable Bytes: 0.
ANR2208I Volume C:\WIN32APP\IBM\ADSM\SERVER\DATA01.DSM deleted from storage pool
BACKUPPOOL.
```

This restores all backup data for the BACKUPPOOL pool onto the new storage pool volumes and deletes the destroyed volumes from the pool when complete. The restore command can then be used to restore the ARCHIVEPOOL disk storage pool.

2. You can now recover the primary tape pools, again using the RESTORE STGPOOL command. As with disk pools, the volumes to be restored must have an access mode of destroyed. Their mode was changed to unavailable when you updated the storage pool volumes to prevent the server from attempting to mount them. For the actual recovery you have to update the access mode again, this time to destroyed:

```
adsm> update volume * acc=destroyed whereaccess=unavailable
ANR2207I Volume LIBR01 updated.
ANR2207I Volume LIBR02 updated.
ANR2116E UPDATE VOLUME: Access mode for volume LIBR03 cannot be changed to
"destroyed" - volume does not belong to a primary storage pool.
```

The error message indicating that LIBR03 cannot be changed is not a problem. This volume belongs to the BACK_COPY pool, which we are not going to recover. Its mode will remain as unavailable.

To restore these volumes we must create new scratch volumes for the storage pools being restored. We use the DSMFMT utility. We placed six new 4MM tapes into the Aiwa library and entered the following command:

```
dsmlabel -drive=mt3,32768 -library=lb2 -search -keep -overwrite
```

This searched the library, labeling each tape and prompting us to enter a label for each tape in turn. For the restore, as with disk volumes, the labels must be different from the labels of the volumes being restored. We used a new sequence of labels: LIBR11 to LIBR16. The -keep

parameter left the tapes in the library to simplify the following CHECKIN LIBVOLUME command:

```
adsm> checkin libvolume 4mmlib status=scratch search=yes
ANR8422I CHECKIN LIBVOLUME: Operation for library 4MMLIB started as process 9

ANR8430I Volume LIBR11 has been checked into library 4MMLIB.
ANR8430I Volume LIBR12 has been checked into library 4MMLIB.
ANR8430I Volume LIBR13 has been checked into library 4MMLIB.
ANR8430I Volume LIBR14 has been checked into library 4MMLIB.
ANR8430I Volume LIBR15 has been checked into library 4MMLIB.
ANR8430I Volume LIBR16 has been checked into library 4MMLIB.
ANR8431I CHECKIN LIBVOLUME process completed for library 4MMLIB; 6 volume(s)
found.
```

This command searches the library and checks in all volumes it finds as scratch volumes. On completion this can be confirmed with the QUERY LIBVOLUME command:

```
adsm> query libvolume

Library Name     Volume Name     Status                Last Use
------------     -----------     -----------------     ---------
4MMLIB           LIBR11          Scratch
4MMLIB           LIBR12          Scratch
4MMLIB           LIBR13          Scratch
4MMLIB           LIBR14          Scratch
4MMLIB           LIBR15          Scratch
4MMLIB           LIBR16          Scratch
```

We now have scratch volumes available for the restore so we can start restoring the primary tape pools, beginning with the BACK_TAPE pool:

```
adsm> restore stgpool back_tape

ANR1230I Restore of primary storage pool BACK_TAPE started as process 15.

ANR1254I Removable volume COPY01 is required for restore processing.
ANR8324I 8MM volume COPY01 is expected to be mounted (R/W).
ANR8326I 008: Mount 8MM volume COPY01 R/W in drive DRIVE1 (MT1.0.0.1) of library
8MMLIB within 60 minutes.
ANR8337I 4MM volume LIBR11 mounted in drive DRIVE0 (MT3).
ANR1340I Scratch volume LIBR11 is now defined in storage pool BACK_TAPE.
ANR8335I 008: Verifying label of 8MM volume COPY01 in drive DRIVE1 (MT1.0.0.1).
ANR8328I 008: 8MM volume COPY01 mounted in drive DRIVE1 (MT1.0.0.1).

ANR1254I Removable volume COPY02 is required for restore processing.
ANR8324I 8MM volume COPY02 is expected to be mounted (R/W).
ANR8336I Verifying label of 8MM volume COPY01 in drive DRIVE1 (MT1.0.0.1).
ANR8468I 8MM volume COPY01 dismounted from drive DRIVE1 (MT1.0.0.1) in library
8MMLIB.
ANR8326I 009: Mount 8MM volume COPY02 R/W in drive DRIVE1 (MT1.0.0.1) of library
8MMLIB within 60 minutes.
ANR8335I 009: Verifying label of 8MM volume COPY02 in drive DRIVE1 (MT1.0.0.1).
ANR8328I 009: 8MM volume COPY02 mounted in drive DRIVE1 (MT1.0.0.1).
ANR1341I Scratch volume LIBR01 has been deleted from storage pool BACK_TAPE
ANR1234I Restore process 15 ended for storage pool BACK_TAPE.
ANR1238I Restore of primary storage pool BACK_TAPE has ended.  Files Restored:
5867, Bytes Restored: 940352621, Unreadable Files: 0, Unreadable Bytes: 0.

ANR1341I Scratch volume LIBR02 has been deleted from storage pool BACK_TAPE
```

As with the disk pool restore, the DR_COPY pool volumes are mounted one after the other until all data has been restored. The restore process acquires a new scratch tape, LIBR11, from the library for the newly restored BACK_TAPE storage pool and deletes destroyed volumes LIBR01 and LIBR02 from the pool. The same process can be repeated for the other primary tape storage pools.

3. Having recovered the storage pools, their contents may be out of sync with the database that was recovered using a point-in-time restore. This can now be resolved by using the AUDIT VOLUME command to audit the individual storage pool volumes.

This concludes the server disaster recovery example. When the server is up and running again, we recommend that you start the normal offsite backup process as soon as possible.

Appendix A. Sample Recovery Command File and Report

This appendix contains an example of a Windows batch file and the associated ADSM macro it invokes to obtain all server information required to perform a complete server recovery. Also included is an example of the report produced by running the command and macro.

A.1 Offsite Command

OFFSITE.CMD is a sample batch file that will create the files needed to recover the server, as described in Chapter 7, “Server Recovery Example” on page 107 and listed in Table 1 on page 114.

OFFSITE.CMD invokes a number of ADSM administrator commands and a macro in batch mode. It contains variables that can be modified to control how it works and where it places its output. To tailor this command file locate the line:

```
REM ************************* Set Up Variables: Start ********
```

Following this line are the variables you can customize. They indicate the ADSM server install directory, the ADSM administrator directory, and the file’s destination directory. The destination directory by default points to drive A:\, but it could be changed to point to a networked drive and directory in a remote location.

```
echo off
REM Sample BAT executable file to create files required
REM for ADSM server recovery.
REM Customize this sample cmd to suit your environment.
REM
REM To execute:
REM
REM  Enter: OFFSITE  admin_id  admin_passwd
REM
REM      where admin_id and admin_passwd are ADSM
REM      administrator id and password
REM
REM ************************* Set Up Variables: Start ********
REM
REM Change s_drv and s_dir to reflect the drive
REM and directory where the ADSM server is installed

set s_drv=C:
set s_dir=\WIN32APP\IBM\ADSM\server
```

```
REM Change a_drv and a_dir to reflect the drive
REM and directory where the ADSM administrator is installed

set a_drv=C:
set a_dir=\WIN32APP\IBM\ADSM\saclient

REM Change d_drv and d_dir to reflect the drive
REM and directory where the destination files are stored

set d_drv=A:
set d_dir=\

REM ************************* Set Up Variables: End **********
REM Check to see if user specified ADSM admin id and password

set nothinghere=
if A%1 == A%nothinghere% goto noparms

REM Set current dir to ADSM server dir
REM where the server is installed

%s_drv%
cd %s_dir%

ECHO Copy the server options file to %d_drv%%d_dir%\DSMSERV.OPT
copy dsmserv.opt %d_drv%%d_dir%\DSMSERV.OPT

REM set path to admin client path

%a_drv%
cd %a_dir%

ECHO Create server reports in %d_drv%%d_dir%\SERVER.REP
dsmadmc -id=%1 -pass=%2 macro %s_drv%%s_dir%\offsite.mac > %d_drv%%d_dir%\server.rep

ECHO Backup the volume history file to %d_drv%%d_dir%\VOLHIST.OUT
dsmadmc -id=%1 -pass=%2 backup volh filename=%d_drv%%d_dir%\VOLHIST.OUT

ECHO Backup the device configuration file to %d_drv%%d_dir%\DEVCNFG.OUT
dsmadmc -id=%1 -pass=%2 backup devconf filename=%d_drv%%d_dir%\DEVCNFG.OUT

REM Change back to server directory
REM where the server is installed
%s_drv%
cd %s_dir%
goto exit
```

```
:noparms
ECHO Please specify ADSM administrator id and password
ECHO  OFFSITE adminid adminpw

:exit
```

A.2 OFFSITE.MAC

OFFSITE.CMD runs a number of ADSM commands and an administrative macro called OFFSITE.MAC:

```
q sta
q opt
q db f=d
q dbv f=d
q log f=d
q logv f=d
query stgpool f=d
q volh type=dbb
q vol stgpool=dr_copy
q vol f=d
query dbb f=d
```

A.3 Sample SERVER.REP

```
ADSTAR Distributed Storage Manager
Command Line Administrative Interface - Version 2, Release 1, Level 0.4
(C) Copyright IBM Corporation, 1990, 1996, All Rights Reserved.

ANS5100I Session established with server ADSM: Windows NT
ANS5101I Server command: 'q sta'
ADSM Server for Windows NT - Version 2, Release 1, Level 0.12/0.12


                     Server Name: ADSM
    Server Installation Date/Time: 11/26/1996 10:36:21
         Server Restart Date/Time: 12/19/1996 18:22:46
                   Authentication: On
       Password Expiration Period: 90 Day(s)
                     Registration: Closed
                     Availability: Enabled
                       Accounting: On
    Activity Log Retention Period: 1 Day(s)
             License Audit Period: 30 Day(s)
               Last License Audit: 11/26/1996 10:38:16
        Server License Compliance: Valid
                Central Scheduler: Active
                 Maximum Sessions: 25
```

```
       Maximum Scheduled Sessions: 12
    Event Record Retention Period: 10 Day(s)
Schedule Randomization Percentage: 25
           Query Schedule Period: Client's Choice
          Maximum Command Retries: Client's Choice
                    Retry Period: Client's Choice
                 Scheduling Modes: Any
                         Log Mode: RollForward
          Database Backup Trigger: Enabled

ANS5101I Server command: 'q opt'

Server Option     Option Setting       Server Option     Option Setting
---------------- -------------------- ---------------- --------------------
CommTimeOut       60                   IdleTimeOut       15
BufPoolSize       2048                 LogPoolSize       512
DateFormat        1 (mm/dd/yyyy)       TimeFormat        1 (hh:mm:ss)
NumberFormat      1 (1,000.00)         MessageFormat     1
Language          AMENG                MaxSessions       25
ExpInterval       24                   ExpQuiet          No
MirrorRead DB     Normal               MirrorRead LOG    Normal
MirrorWrite DB    Sequential           MirrorWrite LOG   Parallel
VolumeHistory     volhist.out          Devconfig         devcnfg.out
TxnGroupMax       40                   MoveBatchSize     40
MoveSizeThresh    500                  StatusMsgCnt      10
TcpPort           1500                 TCPWindowsize     8192
TCPNoDelay        No                   IPXSocket         8522
NetbiosBufferSize 32768                NetbiosSessions   25
IpxBufferSize     32768                NamedPipeName     \\.\PIPE\ADSMPIPE
CommMethod        TCPIP                CommMethod        NAMEDPIPE
Message Interval  1                    MaxMemory         536870912
ANS5101I Server command: 'q db f=d'

         Available Space (MB): 32
       Assigned Capacity (MB): 32
       Maximum Extension (MB): 0
       Maximum Reduction (MB): 24
            Page Size (bytes): 4,096
        Total Usable Pages: 8,192
                Used Pages: 1,113
                     %Util: 13.6
                Max. %Util: 13.6
          Physical Volumes: 1
         Buffer Pool Pages: 512
     Total Buffer Requests: 3,837
           Cache Hit Pct.: 93.43
          Cache Wait Pct.: 0.00
      Backup in Progress?: Yes
```

```
    Type of Backup In Progress: Full
  Incrementals Since Last Full: 0
Changed Since Last Backup (MB): 0.02
            Percentage Changed: 0.36
Last Complete Backup Date/Time: 12/04/1996 16:27:50

ANS5101I Server command: 'q dbv f=d'

Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\DB1.DSM
         Copy Status: Sync'd
Volume Name (Copy 2):
         Copy Status: Undefined
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 32
Allocated Space (MB): 32
     Free Space (MB): 0

ANS5101I Server command: 'q log f=d'

       Available Space (MB): 48
     Assigned Capacity (MB): 48
     Maximum Extension (MB): 0
     Maximum Reduction (MB): 32
          Page Size (bytes): 4,096
         Total Usable Pages: 11,776
                 Used Pages: 2,702
                      %Util: 22.9
                 Max. %Util: 23.0
           Physical Volumes: 1
              Log Pool Pages: 128
         Log Pool Pct. Util: 1.64
         Log Pool Pct. Wait: 0.00
Cumulative Consumption (MB): 73.42
Consumption Reset Date/Time: 12/04/1996 17:34:21

ANS5101I Server command: 'q logv f=d'

Volume Name (Copy 1): C:\WIN32APP\IBM\ADSM\SERVER\LOG1.DSM
         Copy Status: Sync'd
Volume Name (Copy 2):
         Copy Status: Undefined
Volume Name (Copy 3):
         Copy Status: Undefined
Available Space (MB): 48
Allocated Space (MB): 48
     Free Space (MB): 0
```

```
ANS5101I Server command: 'q stg f=d'

               Storage Pool Name: ARCHIVEPOOL
               Storage Pool Type: Primary
               Device Class Name: DISK
         Estimated Capacity (MB): 25.0
                         %Util: 5.1
                         %Migr: 5.1
                      High Mig%: 80
                       Low Mig%: 60
            Migration Processes: 1
              Next Storage Pool: ARCH_TAPE
          Maximum Size Threshold: No Limit
                        Access: Read/Write
                   Description:
           Cache Migrated Files?: Yes
                     Collocate?:
           Reclamation Threshold:
 Maximum Scratch Volumes Allowed:
   Delay Period for Volume Reuse:
          Migration in Progress?: No
            Amount Migrated (MB): 0.00
Elapsed Migration Time (seconds): 0
        Reclamation in Progress?:
 Volume Being Migrated/Reclaimed:
  Last Update by (administrator): TIM
           Last Update Date/Time: 11/26/1996 11:17:53

               Storage Pool Name: ARCH_COPY
               Storage Pool Type: Copy
               Device Class Name: 4MM
         Estimated Capacity (MB): 0.0
                         %Util: 0.0
                         %Migr:
                      High Mig%:
                       Low Mig%:
            Migration Processes:
              Next Storage Pool:
          Maximum Size Threshold:
                        Access: Read/Write
                   Description:
           Cache Migrated Files?:
                     Collocate?: No
           Reclamation Threshold: 100
 Maximum Scratch Volumes Allowed: 100
   Delay Period for Volume Reuse: 0 Day(s)
          Migration in Progress?:
```

```
           Amount Migrated (MB):
Elapsed Migration Time (seconds):
        Reclamation in Progress?: No
 Volume Being Migrated/Reclaimed:
  Last Update by (administrator): TIM
           Last Update Date/Time: 11/26/1996 13:03:42

               Storage Pool Name: ARCH_TAPE
               Storage Pool Type: Primary
               Device Class Name: 4MM
         Estimated Capacity (MB): 0.0
                           %Util: 0.0
                           %Migr: 0.0
                       High Mig%: 90
                        Low Mig%: 70
             Migration Processes:
               Next Storage Pool:
          Maximum Size Threshold: No Limit
                          Access: Read/Write
                     Description:
           Cache Migrated Files?:
                       Collocate?: No
           Reclamation Threshold: 100
 Maximum Scratch Volumes Allowed: 100
   Delay Period for Volume Reuse: 0 Day(s)
         Migration in Progress?: No
           Amount Migrated (MB): 0.00
Elapsed Migration Time (seconds): 0
        Reclamation in Progress?: No
 Volume Being Migrated/Reclaimed:
  Last Update by (administrator): TIM
           Last Update Date/Time: 11/26/1996 11:16:50

               Storage Pool Name: BACKUPPOOL
               Storage Pool Type: Primary
               Device Class Name: DISK
         Estimated Capacity (MB): 25.0
                           %Util: 8.3
                           %Migr: 8.3
                       High Mig%: 80
                        Low Mig%: 60
             Migration Processes: 1
               Next Storage Pool: BACK_TAPE
          Maximum Size Threshold: No Limit
                          Access: Read/Write
                     Description:
           Cache Migrated Files?: Yes
                       Collocate?:
```

```
          Reclamation Threshold:
 Maximum Scratch Volumes Allowed:
   Delay Period for Volume Reuse:
         Migration in Progress?: No
           Amount Migrated (MB): 0.00
Elapsed Migration Time (seconds): 0
        Reclamation in Progress?:
 Volume Being Migrated/Reclaimed:
  Last Update by (administrator): TIM
           Last Update Date/Time: 11/26/1996 11:17:38

               Storage Pool Name: BACK_COPY
               Storage Pool Type: Copy
               Device Class Name: 4MM
         Estimated Capacity (MB): 381,400.0
                         %Util: 0.1
                         %Migr:
                      High Mig%:
                       Low Mig%:
            Migration Processes:
              Next Storage Pool:
          Maximum Size Threshold:
                        Access: Read/Write
                   Description:
           Cache Migrated Files?:
                     Collocate?: No
           Reclamation Threshold: 100
 Maximum Scratch Volumes Allowed: 100
   Delay Period for Volume Reuse: 0 Day(s)
         Migration in Progress?:
           Amount Migrated (MB):
Elapsed Migration Time (seconds):
        Reclamation in Progress?: No
 Volume Being Migrated/Reclaimed:
  Last Update by (administrator): TIM
           Last Update Date/Time: 11/26/1996 13:03:52

               Storage Pool Name: BACK_TAPE
               Storage Pool Type: Primary
               Device Class Name: 4MM
         Estimated Capacity (MB): 381,400.0
                         %Util: 0.2
                         %Migr: 1.0
                      High Mig%: 90
                       Low Mig%: 70
            Migration Processes:
              Next Storage Pool:
          Maximum Size Threshold: No Limit
```

```
                        Access: Read/Write
                   Description:
          Cache Migrated Files?:
                     Collocate?: No
          Reclamation Threshold: 60
 Maximum Scratch Volumes Allowed: 100
   Delay Period for Volume Reuse: 0 Day(s)
          Migration in Progress?: No
            Amount Migrated (MB): 0.00
Elapsed Migration Time (seconds): 0
        Reclamation in Progress?: No
 Volume Being Migrated/Reclaimed:
  Last Update by (administrator): TIM
           Last Update Date/Time: 11/26/1996 11:16:19

               Storage Pool Name: DR_COPY
               Storage Pool Type: Copy
               Device Class Name: 8MM
         Estimated Capacity (MB): 494,400.0
                           %Util: 0.2
                           %Migr:
                       High Mig%:
                        Low Mig%:
             Migration Processes:
               Next Storage Pool:
          Maximum Size Threshold:
                        Access: Read/Write
                   Description:
          Cache Migrated Files?:
                     Collocate?: No
          Reclamation Threshold: 100
 Maximum Scratch Volumes Allowed: 100
   Delay Period for Volume Reuse: 35 Day(s)
          Migration in Progress?:
            Amount Migrated (MB):
Elapsed Migration Time (seconds):
        Reclamation in Progress?: No
 Volume Being Migrated/Reclaimed:
  Last Update by (administrator): TIM
           Last Update Date/Time: 11/26/1996 13:40:44

               Storage Pool Name: SPACEMGPOOL
               Storage Pool Type: Primary
               Device Class Name: DISK
         Estimated Capacity (MB): 25.0
                           %Util: 0.0
                           %Migr: 0.0
                       High Mig%: 90
```

```
                     Low Mig%: 70
            Migration Processes: 1
              Next Storage Pool:
          Maximum Size Threshold: No Limit
                      Access: Read/Write
                 Description:
           Cache Migrated Files?: Yes
                    Collocate?:
           Reclamation Threshold:
 Maximum Scratch Volumes Allowed:
   Delay Period for Volume Reuse:
          Migration in Progress?: No
            Amount Migrated (MB): 0.00
Elapsed Migration Time (seconds): 0
        Reclamation in Progress?:
 Volume Being Migrated/Reclaimed:
  Last Update by (administrator): SERVER_CONSOLE
           Last Update Date/Time: 11/26/1996 10:38:16

ANS5101I Server command: 'q volh type=dbb'

       Date/Time: 11/26/1996 12:47:14
     Volume Type: BACKUPFULL
   Backup Series: 1
Backup Operation: 0
      Volume Seq: 1
    Device Class: ZIP
     Volume Name: ZIP001
 Volume Location:

       Date/Time: 11/26/1996 13:54:39
     Volume Type: BACKUPINCR
   Backup Series: 1
Backup Operation: 1
      Volume Seq: 1
    Device Class: ZIP
     Volume Name: ZIP002
 Volume Location:

       Date/Time: 12/04/1996 16:27:50
     Volume Type: BACKUPINCR
   Backup Series: 1
Backup Operation: 2
      Volume Seq: 1
    Device Class: ZIP
     Volume Name: ZIP003
 Volume Location:
```

```
       Date/Time: 12/04/1996 16:54:10
     Volume Type: BACKUPINCR
   Backup Series: 1
Backup Operation: 3
      Volume Seq: 1
    Device Class: ZIP
     Volume Name: ZIP004
 Volume Location:

ANS5101I Server command: 'q vol stgpool=dr_copy'

Volume Name               Storage      Device      Estimated  %Util   Volume
                          Pool Name    Class Name   Capacity          Status
                                                       (MB)
------------------------  -----------  ----------  ---------  -----  --------
COPY01                    DR_COPY      8MM           4,944.0   10.4  Filling
COPY02                    DR_COPY      8MM           4,944.0    7.8  Filling
ANS5101I Server command: 'q vol f=d'

                  Volume Name: LIBR03
             Storage Pool Name: BACK_COPY
             Device Class Name: 4MM
       Estimated Capacity (MB): 3,814.0
                       %Util: 13.3
                Volume Status: Filling
                       Access: Unavailable
        Pct. Reclaimable Space: 0.4
                Scratch Volume?: Yes
                In Error State?: No
      Number of Writable Sides: 1
       Number of Times Mounted: 10
             Write Pass Number: 1
     Approx. Date Last Written: 11/27/1996 15:42:20
        Approx. Date Last Read: 11/27/1996 16:11:41
           Date Became Pending:
        Number of Write Errors: 0
         Number of Read Errors: 0
               Volume Location:
Last Update by (administrator): SERVER_CONSOLE
         Last Update Date/Time: 12/04/1996 17:56:45

                  Volume Name: LIBR01
             Storage Pool Name: BACK_TAPE
             Device Class Name: 4MM
       Estimated Capacity (MB): 3,814.0
                       %Util: 23.5
                Volume Status: Filling
                       Access: Read/Write
```

```
        Pct. Reclaimable Space: 0.0
               Scratch Volume?: Yes
               In Error State?: No
      Number of Writable Sides: 1
       Number of Times Mounted: 2
             Write Pass Number: 1
     Approx. Date Last Written: 12/06/1996 12:27:29
        Approx. Date Last Read: 12/13/1996 09:48:57
           Date Became Pending:
        Number of Write Errors: 0
         Number of Read Errors: 0
               Volume Location:
Last Update by (administrator):
         Last Update Date/Time: 12/06/1996 11:37:42

                   Volume Name: LIBR02
               Storage Pool Name: BACK_COPY
               Device Class Name: 4MM
       Estimated Capacity (MB): 3,814.0
                        %Util: 0.1
                  Volume Status: Filling
                         Access: Read/Write
        Pct. Reclaimable Space: 0.0
               Scratch Volume?: Yes
               In Error State?: No
      Number of Writable Sides: 1
       Number of Times Mounted: 2
             Write Pass Number: 1
     Approx. Date Last Written: 12/13/1996 15:15:13
        Approx. Date Last Read: 12/13/1996 15:15:13
           Date Became Pending:
        Number of Write Errors: 0
         Number of Read Errors: 0
               Volume Location:
Last Update by (administrator):
         Last Update Date/Time: 12/13/1996 15:15:02

                   Volume Name: C:\WIN32APP\IBM\ADSM\SERVER\ARCH01.DSM
               Storage Pool Name: ARCHIVEPOOL
               Device Class Name: DISK
       Estimated Capacity (MB): 25.0
                        %Util: 5.1
                  Volume Status: On-Line
                         Access: Read/Write
        Pct. Reclaimable Space:
               Scratch Volume?:
               In Error State?:
      Number of Writable Sides:
```

```
       Number of Times Mounted:
             Write Pass Number:
     Approx. Date Last Written:
        Approx. Date Last Read:
           Date Became Pending:
        Number of Write Errors:
         Number of Read Errors:
               Volume Location:
Last Update by (administrator): SERVER_CONSOLE
         Last Update Date/Time: 12/06/1996 10:01:15

                     Volume Name: C:\WIN32APP\IBM\ADSM\SERVER\BACK01.DSM
               Storage Pool Name: BACKUPPOOL
               Device Class Name: DISK
       Estimated Capacity (MB): 25.0
                           %Util: 8.3
                  Volume Status: On-Line
                         Access: Read/Write
        Pct. Reclaimable Space:
                Scratch Volume?:
                In Error State?:
      Number of Writable Sides:
       Number of Times Mounted:
             Write Pass Number:
     Approx. Date Last Written:
        Approx. Date Last Read:
           Date Became Pending:
        Number of Write Errors:
         Number of Read Errors:
               Volume Location:
Last Update by (administrator): SERVER_CONSOLE
         Last Update Date/Time: 12/06/1996 10:00:55

                     Volume Name: C:\WIN32APP\IBM\ADSM\SERVER\HSM01.DSM
               Storage Pool Name: SPACEMGPOOL
               Device Class Name: DISK
       Estimated Capacity (MB): 25.0
                           %Util: 0.0
                  Volume Status: On-Line
                         Access: Read/Write
        Pct. Reclaimable Space:
                Scratch Volume?:
                In Error State?:
      Number of Writable Sides:
       Number of Times Mounted:
             Write Pass Number:
     Approx. Date Last Written:
        Approx. Date Last Read:
```

```
           Date Became Pending:
         Number of Write Errors:
          Number of Read Errors:
                Volume Location:
Last Update by (administrator): SERVER_CONSOLE
         Last Update Date/Time: 12/06/1996 10:01:43

                    Volume Name: COPY01
               Storage Pool Name: DR_COPY
               Device Class Name: 8MM
        Estimated Capacity (MB): 4,944.0
                         %Util: 10.4
                  Volume Status: Filling
                         Access: Read/Write
         Pct. Reclaimable Space: 0.3
                Scratch Volume?: Yes
                In Error State?: No
       Number of Writable Sides: 1
        Number of Times Mounted: 9
              Write Pass Number: 1
     Approx. Date Last Written: 11/27/1996 16:48:18
        Approx. Date Last Read: 12/06/1996 11:38:07
           Date Became Pending:
         Number of Write Errors: 0
          Number of Read Errors: 0
                Volume Location: tims vault
Last Update by (administrator): SERVER_CONSOLE
         Last Update Date/Time: 12/04/1996 18:08:21

                    Volume Name: COPY02
               Storage Pool Name: DR_COPY
               Device Class Name: 8MM
        Estimated Capacity (MB): 4,944.0
                         %Util: 7.8
                  Volume Status: Filling
                         Access: Read/Write
         Pct. Reclaimable Space: 0.0
                Scratch Volume?: Yes
                In Error State?: No
       Number of Writable Sides: 1
        Number of Times Mounted: 11
              Write Pass Number: 1
     Approx. Date Last Written: 12/04/1996 16:47:33
        Approx. Date Last Read: 12/06/1996 12:08:53
           Date Became Pending:
         Number of Write Errors: 0
          Number of Read Errors: 0
```

```
              Volume Location: tims vault
Last Update by (administrator): SERVER_CONSOLE
         Last Update Date/Time: 12/04/1996 18:08:21

ANS5101I Server command: 'q dbb f=d'

             Full Device Class: ZIP
      Incremental Device Class: ZIP
           Log Full Percentage: 85
    Incrementals Between Fulls: 0
Last Update by (administrator): TIM
         Last Update Date/Time: 12/19/1996 18:30:14


ANS5103I Highest return code was 0.
```

Appendix B. Special Notices

This publication is intended to help ADSM business partners, customers, consultants, and IBM personnel configure and perform recovery of the ADSM server for Windows NT. The information in this publication is not intended as the specification of any programming interfaces that are provided by ADSM for Windows NT. See the PUBLICATIONS section of the IBM Programming Announcement for ADSM for Windows NT for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain 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.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ADSTAR	AIX
BookManager	IBM
OS/2	PowerPC
PS/2	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo
are trademarks or registered trademarks of Microsoft Corporation.

AIWA	AIWA, Inc.
DLT	Quantum Corporation
Exabyte	Exabyte Corporation
Intel	Intel Corporation
IPX/SPX	Novell, Inc.
Jaz	Iomega Corporation
NetWare	Novell, Inc.
NFS	Sun Microsystems, Inc.
Pentium	Intel Corporation
Windows 95	Microsoft Corporation
Windows NT	Microsoft Corporation
Zip	Iomega Corporation

Other trademarks are trademarks of their respective companies.

Appendix C. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 151.

General Topics

- *ADSM Presentation Guide*, GG24-4146
- *ADSM Version 2 Presentation Guide*, SG24-4532
- *ADSM Advanced Implementation Experiences*, GG24-4221
- *Using ADSM Hierarchical Storage Management*, SG24-4631

Specific Server Books

- *Getting Started with ADSM/6000*, GG24-4421
- *ADSM for AIX: Advanced Topics*, SG24-4601
- *AIX Tape Management*, SG24-4705
- *ADSM/6000 on 9076 SP2*, GG24-4499
- *ADSM for MVS: Recovery and Disaster Recovery*, SG24-4537
- *ADSM for MVS: Using Tapes and Tape Libraries*, SG24-4538
- *Getting Started with ADSM/2*, GG24-4321
- *ADSM for OS/2: Advanced Topics*, SG24-4740
- *Setting Up and Implementing ADSM/400*, GG24-4460
- *ADSM/VSE Implementation Guide*, SG24-4266

Specific Client Books

- *Getting Started with ADSM NetWare Clients*, GG24-4242
- *Getting Started with ADSM AIX Clients*, GG24-4243
- *ADSM API Examples for OS/2 and Windows*, SG24-2588

ADSM with Other Products

- *Using ADSM to Back Up Databases*, SG24-4335

- *Using ADSM to Back Up Lotus Notes*, SG24-4534
- *HSM for NetWare: ADSM and AvailHSM Implementation*, SG24-4713
- *Using ADSM to Back Up OS/2 LAN Server & Warp Server*, SG24-4682
- *Backup, Recovery, and Availability with DB2 PE*, SG24-4695

Coming Soon

- *Client Disaster Recovery: Bare Metal Restore*, SG24-4880
- *ADSM Technical Overview*, SG24-4877

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

C.3 Other Publications

These publications are also relevant as further information sources:

- ADSM Publications
  - *ADSM Online Product Library CD-ROM*, SK2T-8714

    All of the ADSM publications are available in online readable format on the CD-ROM listed above. The ADSM library is also available on the following CD-ROMs:

    - *MVS Base Collection Kit*, SK2T-0710
    - *VM Base Collection Kit*, SK2T-2067
    - *AS/400 Base Collection Kit*, SK2T-2171
    - *IBM SystemView for AIX*, SK2T-1451
  - *ADSM V2R1 General Information*, GH35-0131
  - *ADSM V2R1 Windows NT License*, GC35-0234

- *ADSM V2R1 Windows NT Quick Start*, GC35-0235
- *ADSM V2R1 Windows NT Admin Guide*, GC35-0236
- *ADSM V2R1 Windows NT Admin. Ref.*, GC35-0237

- *ADSM V2R1 Installing the Clients*, SH26-4049-02
- *ADSM V2R1 Using the UNIX HSM Clients*, SH26-4030
- *ADSM V2R1 Using the UNIX Backup-Archive Client*, SH26-4052
- *ADSM V2R1 Using the OS/2 Backup-Archive Client*, SH26-4053
- *ADSM V2R1 Using the DOS Backup-Archive Client*, SH26-4054
- *ADSM V2R1 Using the Microsoft Windows Backup-Archive Client*, SH26-4056
- *ADSM V2R1 Using the Novell NetWare Backup-Archive Client*, SH26-4055
- *ADSM V2R1 Using the Apple Macintosh Backup-Archive Client*, SH26-4051
- *ADSM V2R1 Using the Lotus Notes Backup Agent*, SH26-4047
- *ADSM V2R1 Reference Cards for Backup-Archive Clients*, SX26-6013

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com/redbooks.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet**

  Type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```
- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks/redbooks.html
- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.
- **ITSO4USA category on INEWS**
- **Online** — send orders to:

  USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword

subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet)

  IBMMAIL — send orders to:

In United States:	usib6fpl at ibmmail
In Canada:	caibmbkz at ibmmail
Outside North America:	bookshop at dkibmbsh at ibmmail

  Internet — send orders to:

In United States:	usib6fpl@ibmmail.com
In Canada:	lmannix@vnet.ibm.com
Outside North America:	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29554 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada (toll free)	1-800-267-4455
Outside North America (long distance charge)	(+45) 48 14 2207

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)**

  Ask for:
  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services**

  Send note to softwareshop@vnet.ibm.com

- **Redbooks Home Page on the World Wide Web**

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

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number

- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

Printed in U.S.A.