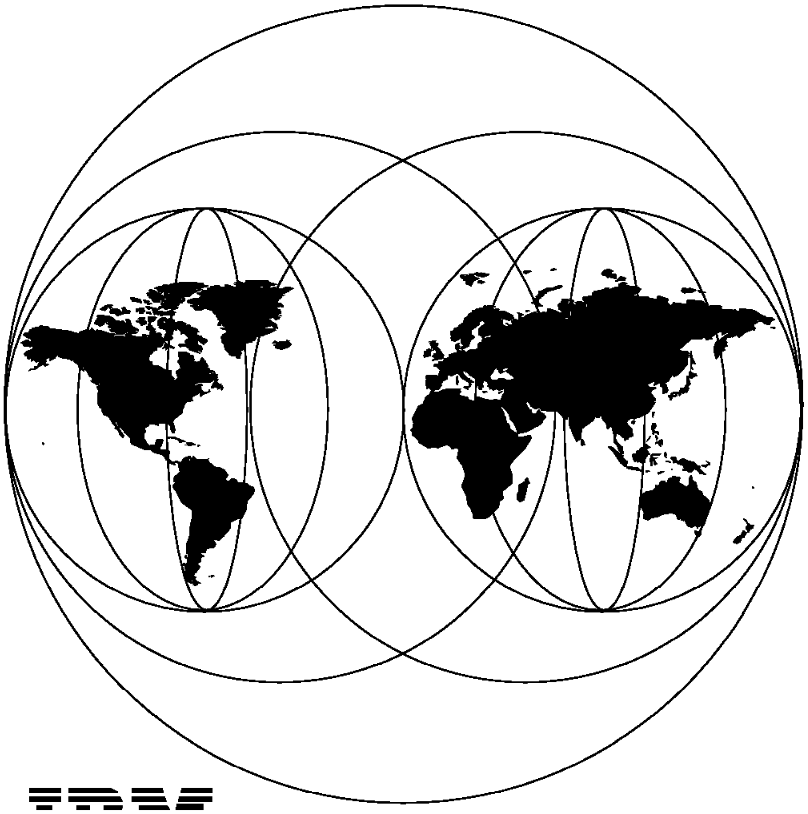
International Technical Support Organization GG24-4499-00

ADSTAR Distributed Storage Manager/6000 on 9076 SP2

March 1995



IBM

International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

GG24-4499-00

ADSTAR Distributed Storage Manager/6000 on 9076 SP2

March 1995

Take Note!

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

First Edition (March 1995)

This edition applies to Version 1 Release 2.0 of ADSTAR Distributed Storage Manager/6000 for use with the AIX/6000 operating system on 9076 SP2.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader′s feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 541 Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

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.

Abstract

This document focuses on ADSM/6000 as the primary IBM program product devoted to file backup and archive in distributed environments and POWERparallel systems. It is unique in its detailed coverage of installation, customization and administration of ADSM/6000 on the IBM POWERparallel Systems SP2. A chapter describes storage management issues adapted to two scenarios: scientific and technical center, and commercial applications. Also, this book provides information about the following topics:

- Administration and operations
- User access issues
- ADSM availability

At last, several script samples are offered to administrators to simplify administrator tasks.

This document was written for IBM system engineers, and storage administrators who have to define a storage management policy and customize it on 9076 SP2. Some knowledge of the AIX/6000 operating system, the IBM AIX Parallel System Support Programs (PSSP), and the 9076 SP2 hardware is assumed.

(164 pages)

Contents

Figures

Tables

Special Notices

This publication is intended to help administrators and IBM system engineers to customize and administer a hierarchical file management environment with ADSTAR Distributed Storage Manager/6000 (ADSM/6000) on POWERparallel systems.

The information in this publication is not intended as the specification of any programming interfaces that are provided by ADSM/6000. See the PUBLICATIONS section of the IBM Programming Announcement for ADSM/6000 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 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
AIX/6000	AIXwindows

AS/400
ESCON
HACMP/6000
IBM
LoadLeveler
Micro Channel
OS/2
POWER Architecture
POWERparallel
POWER2 Architecture
RISC System/6000
RS/6000
Scalable POWERparallel Systems
SP1
SP2
9076 SP1
9076 SP2

The following terms are trademarks of other companies:

DEC, Ultrix	Digital Equipment Corporation
Macintosh	Apple Computer, Incorporated
Microsoft, Windows	Microsoft Corporation
Network File System, NFS	Sun Microsystems, Incorporated
Network Information System, NIS	Sun Microsystems, Incorporated
Novell, Netware	Novell, Incorporated
Solaris	Sun Microsystems, Incorporated
UNIX	X/Open Company Limited

Other trademarks are trademarks of their respective companies.

Preface

This document is for IBM system engineers and customer file management administrators who have to define and customize a hierarchical file management policy for POWERparallel systems. It contains information about the ADSM/6000 IBM primary program product.

This document presents the results experienced by a residency conducted at the ITSO Poughkeepsie center.

How This Document is Organized

The document is organized as follows:

- Chapter 1, "Introduction" is an introduction to storage management issues, to ADSTAR Storage Manager/6000, and to IBM 9076 SP2 hardware and software.
- Chapter 2, "ADSM Installation and Basic Customization" presents the installation and the basic customization of ADSM/6000 on 9076 SP2.
- Chapter 3, "Scenarios" describes typical scientific and commercial environments chosen to discuss particular ADSM/6000 customization scenarios.
- Chapter 4, "Administration and Operations" provides information about the ADSM/6000 administration on 9076 SP2, and presents several topics related to ADSM/6000.
- Chapter 5, "User Access Issues" discusses ADSM/6000 topics involved with the user access of data in an ADSM/6000 environment.
- Chapter 6, "ADSM Availability" includes a presentation and some comments about ADSM/6000 availability.
- Appendix A, ADSM Configuration Files gives examples of client options files.
- Appendix B, The adsmc Interface presents two scripts that are offered to replace the command line interface of ADSM/6000.
- Appendix C, SP2 Users Authorization provides the script written to authorize a user to access data from any node on the IBM 9076 SP2.
- Appendix D, Device Support gives information about the devices that are currently supported by ADSM/6000 on the IBM 9076 SP2.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *ADSTAR Distributed Storage Manager: General Information*, GH35-0114
- *ADSTAR Distributed Storage Manager/6000: Administrator's Guide*, SH26-4005
- *ADSTAR Distributed Storage Manager/6000: Administrator's Reference*, SH26-4006

- *ADSTAR Distributed Storage Manager/6000: Installing the Server and Administrative Client*, SH26-4013
- *ADSTAR Distributed Storage Manager Performance Tuning Guide*, SH26-4034

International Technical Support Organization Publications

- *ADSM Presentation Guide*, GG24-4146
- *ADSM Implementation Examples*, GG24-4034
- *ADSM Advanced Implementation Examples*, GG24-4221
- *Getting Started with ADSM/6000*, GG24-4421
- *Getting Started with ADSM AIX Clients*, GG24-4243
- *Using ADSM to Back Up Databases*, GG24-4335
- *AIX Storage Management*, GG24-4484

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in:

> *International Technical Support Organization Bibliography of Redbooks,* GG24-3070.

To get a catalog of ITSO technical publications (known as "redbooks"), VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

How to Order ITSO Technical Publications

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their local IBM office.

Customers may order hardcopy ITSO books individually or in customized sets, called GBOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain books on a variety of products.

Acknowledgments

This project was designed and managed by:

Hung C. Nguyen
International Technical Support Organization, Poughkeepsie Center

Michel Perraud
International Technical Support Organization, Poughkeepsie Center

The authors of this document are:

Christine O′Sullivan
IBM France

Peter Koepp
IBM Germany

Alfred Leung
IBM Australia

Francesco Pedulla′
IBM Italy

This publication is the result of a residency conducted at the International Technical Support Organization, Poughkeepsie Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Cynthia Behrens
International Technical Support Organization, San Jose Center

Robert J. Curran
IBM Kingston

Jose L. Iglesias
IBM San Jose

Laurence W. Holley
IBM San Jose

Chapter 1. Introduction

This chapter provides a brief overview of storage management functions, ADSM in general, IBM POWERparallel Systems SP2, and running ADSM/6000 on the IBM POWERparallel Systems SP2. It is intended to recall or describe the most relevant issues of ADSM software and the SP2 system and is not meant to be a complete guide.

Section 1. Storage Management Overview

In any computer system, there is only a limited amount of main memory. Main memory is usually volatile, with a few exceptions, such as the non-volatile storage on 3990 control units, with an automatic switch to battery power supply in case of power failure.

On IBM POWERparallel Systems SP2, the main memory of processor nodes is volatile. Therefore, the permanent data of users and applications must be stored on storage media, such as magnetic or optical disks, or magnetic tapes.

The storage becomes constrained when the number of users and applications together with the average size of files is growing. So, storage administrators have more and more difficulties optimizing the use of storage media. The optimization of the use of storage media generally requires a hierarchy of external storage media according to costs and performance.

Moreover, storage administrators are also expected to fulfil the requirements of backups and disaster recovery, which generally implies that permanent data be copied on movable media. These copies are moved to another location and used to restore data in case of accidental destruction.

It may be easy to manage if the storage requirements of your organization are not high, for example, just a few hundred megabytes and there are just a few users on the systems. As the storage requirements continue to grow and more users are added to the system, storage management becomes a complex art.

Consider that you have a computer system with 300GB of disk space supporting 200 users. It will be a very challenging task for a system manager to ensure adequate disk storage is available for end users and at the same time, provide a proper backup and recovery strategy for end users.

There are many high level applications available to provide such services to both system managers and end users. These applications categorize storage devices in terms of their basic properties and provide automatic mechanisms to utilize them in a more efficient way. Such applications are often called *hierarchical storage management* applications. ADSTAR Distribution Storage Manager/6000 (ADSM/6000) is one such application.

1.1 Hierarchical Storage Management

The hierarchical storage management application provides a centralized, policy-driven solution to storage management. It allows the systems manager to specify storage management rules, and the tasks are automatically performed based on those specified rules.

A policy-driven implementation makes storage management easier and relieves the highly-skilled storage administrator of the burden of maintaining batch jobs. High level storage management products commonly use a database for keeping all storage management related data.

The main functions offered by these products are:

1. **Backup/Restore**

The ability to create copies of a client system′s vital data, so that in the event of a failure, the client can be restored to the same state that it was at the time of the last backup.

2. **Archive/Retrieve**

   The ability to free up disk space by moving or archiving infrequently used information to the archive storage devices. If the information is required again, it can be retrieved from the archive.

3. **Migration**

   The ability to structure the storage systems in such a way that each storage device is used in its most efficient way. For example, frequently accessed information, or information requiring high performance access should be stored in fast storage devices (usually disk). Information that is less frequently accessed should be moved to less expensive and often lower performing storage devices, which have usually higher capacity. The flow of data between the devices should happen automatically. This function is the basic mechanism of an automatic disk space management and is provided by ADSM/6000 Version 1 release 2 for the space management of disk storage pools.

4. **Automation**

   The ability to back up, archive, migrate and administer a storage management system without human intervention.

5. **Space Management**

   A mechanism by which the disk space of a client system can be utilized more efficiently. This is done by using the client disk as a cache and maintaining the full information space at a storage server. When data is requested by an application on the client, it is transparently copied to the client cache. The client sees the cache to be as large as the information space at the server.

   This function is not included in ADSM/6000 Version 1 Release 2.

6. **Central Data Management**

   The ability to manage the above mentioned functions from a central point, thereby minimizing the efforts and maximizing efficiency.

1.2 Multilevel Hierarchy

One reason for hierarchical storage management is to make effective use of storage devices and provide automatic mechanisms to use them more efficiently. Effective use of the hierarchical storage management software will result in significant disk savings while providing acceptable access times throughout the hierarchy.

A system manager can define many levels of hierarchy. Usually, there are three levels for hierarchical storage management. Actively used data or information that requires high performance access, such as databases, should reside on level 0 devices (disk), which is usually the fastest, most reliable, but also the most expensive storage in the hierarchy. As access to this data decreases, the information could be moved to level 1 devices, such as optical disk, or level 2 devices like tapes.

Level 1 in the storage hierarchy serves an important purpose by providing an intermediate step between fast and expensive disk devices and high capacity, slower and less expensive tape devices. Aging information or data that is not being accessed frequently can be stored on lower performance disk media, such as optical devices of the hierarchy.

Using a three-level hierarchy approach, data stays on the appropriate storage devices only for the time it is required, thus avoiding consumption of unused storage resource. One can establish policies that provide acceptable response times for level 1 data, and achieve substantial space savings and reduction in disk expense.

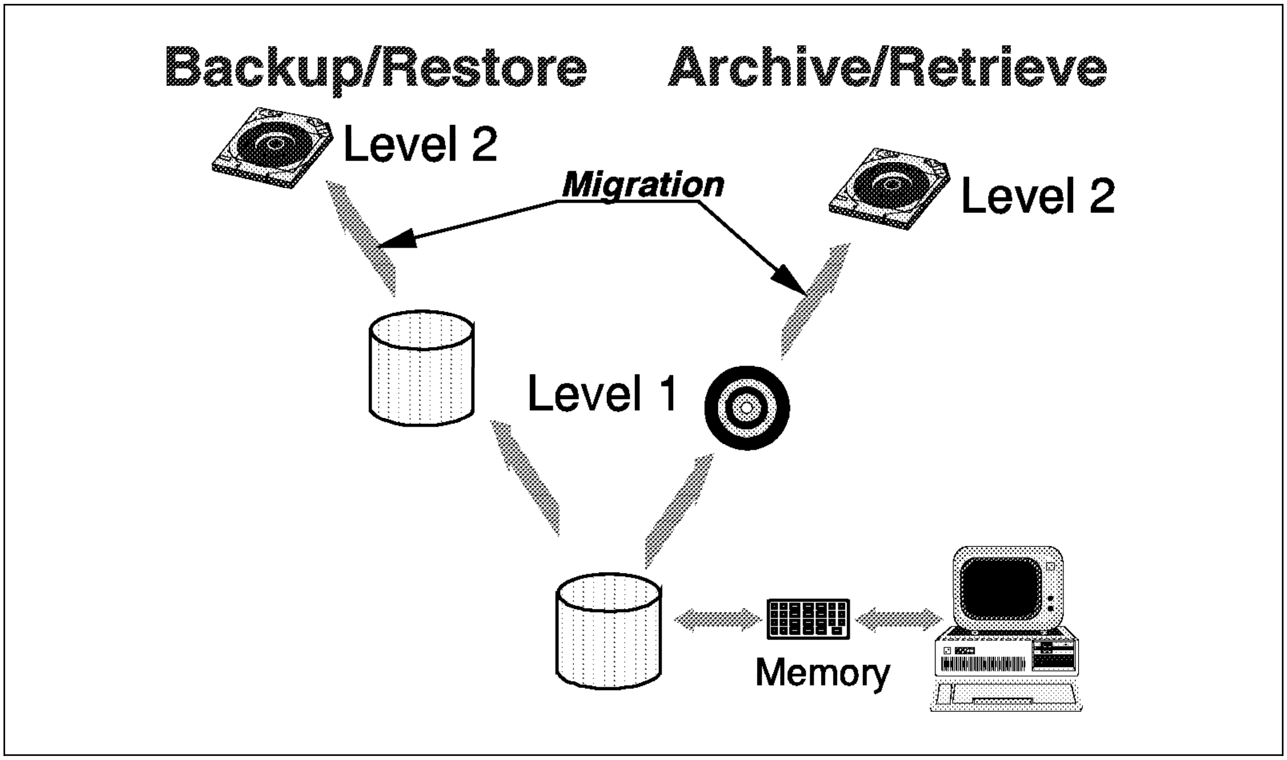


Figure 1. Hierarchical File Management

1.2.1 Storage Devices

As mentioned before, hierarchical storage management is to treat various available storage devices in their most efficient way. This implies that a storage manager has to define a storage hierarchy through which data can be stored and migrated based on space and user requirements. To be able to define a good performing hierarchy, it is necessary to understand the characteristics of storage devices. An extensive guide to such information is available as a redbook (*AIX Storage Management*). Let us look at some of the device characteristics.

1.2.1.1 Disk Storage

Disk storage is generally nonremovable. It allows high read/write speeds and has good capacity capabilities. Single units are now capable of storing up to several gigabytes and with technology improving constantly, there will be higher capacity volumes available soon. Disk also allows direct access to information, and coupled with the capacity and performance, makes it an ideal device for loading information to, and saving from main memory.

There are models of disk drives that do allow the removal of the fixed internal disks. The removed disk packs can then be transferred to and used by other computers. However, due to the proprietary nature of most disk information organization, compatibility is usually only ensured between computers of the same model.

Disk devices constitute the most vital element of the storage system. There are in continuous use as extension of main memory (paging) and as storage for frequently accessed data.

1.2.1.2 Tape Storage

Tape storage technology has been in existence for a long time. The main difference between tape and disk storage devices is that information is read and written sequentially on tapes. This means that access to information on a tape is slow, as the tape must be sequentially moved to the position where the data in search is located. The read/write speed can be very fast, and the capacity is very high, 10GB per tape and more, which means that tapes are currently best suited for backing up large amounts of data, or for infrequent access to archived information.

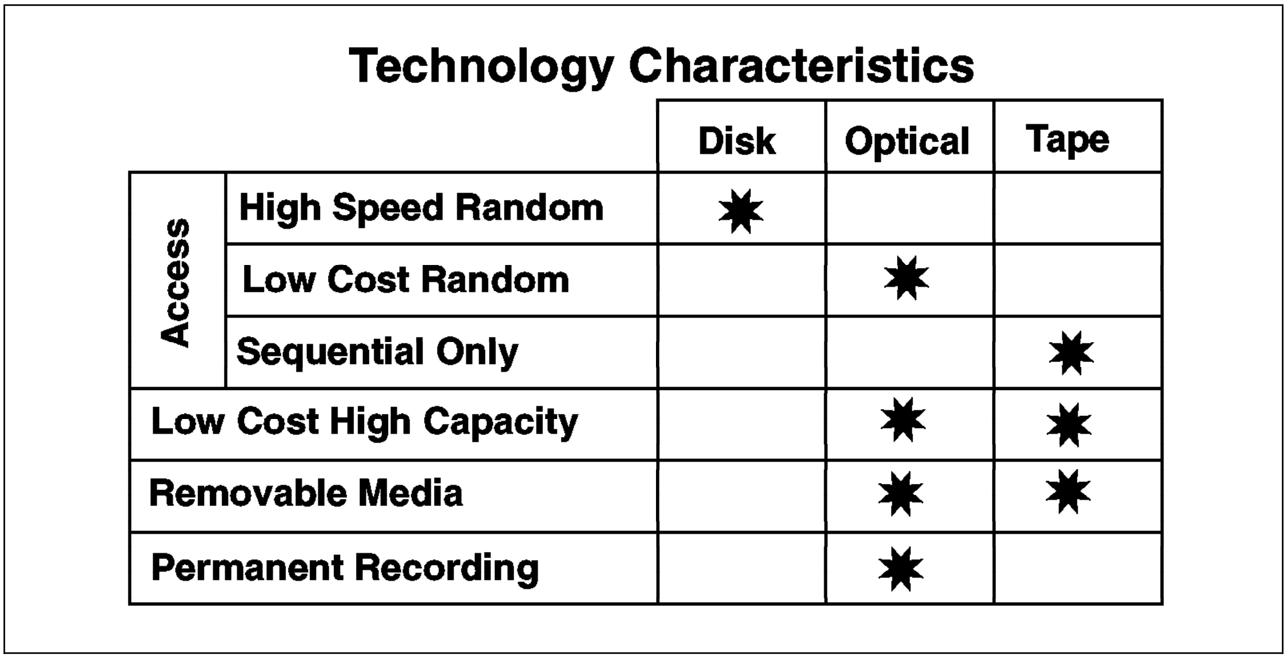
Some tape subsystems provide autochargers with access to a library of tapes. In these instances, tape selection and load/unload time will affect the performance.

1.2.1.3 Optical Storage

A recent technology, optical storage, is based on compact disc technology, though there are several different mechanisms and formats in use. The simplest uses standard CD technology providing a read-only memory (ROM) capability, known as CD-ROM. Later, more complex evolutions allow the disk to be written to once and then read from as a normal CD-ROM; this is known as write-once-read-many (WORM). The latest products offer complete read write capabilities.

Optical storage allows direct access to information, though the access time is somewhat slower than for disk. The capacities range from 640MB to several GB. The extraordinarily long life of information recorded on optical media also makes this a good medium for archive. Similar to tape libraries, optical devices can be packaged into an optical library with capacities ranging from a few dozen GB up to several hundred GBs.

Figure 2 on page 7 summarizes the advantages of each of the device types.



Technology Characteristics

		Disk	Optical	Tape
Access	High Speed Random	✸		
	Low Cost Random		✸	
	Sequential Only			✸
Low Cost High Capacity			✸	✸
Removable Media			✸	✸
Permanent Recording			✸	

Figure 2. Summary of Device Attributes

Section 2. ADSM Overview

ADSTAR Distributed Storage Manager (ADSM) is a client/server product that provides storage management to customers in a wide variety of IBM and non-IBM clients and servers.

ADSTAR Distributed Storage Manager (ADSM) is IBM′s solution to enterprise-wide distributed storage management. It provides highly automated, centrally scheduled, network-based backup and archive functions for workstations and LAN file servers.

ADSM supports a wide variety of platforms as servers and clients. Figure 3 on page 10 shows the ADSM/6000 clients that can be used with the ADSM/6000 server and also the available protocols. Other available ADSM server platforms are MVS, VM, VSE, AS/400, and OS/2. Also, IBM announced ADSM client/server functions for HP-UX and Sun Solaris platforms (planned availability: April 95).

This section introduces ADSM and describes:

- ADSM vocabulary
- ADSM components
- ADSM main functions

2.1 ADSM Vocabulary

This topic discusses vocabulary used generally in ADSM and particularly in ADSM customization.

2.1.1 Storage Pool

A storage pool is a named collection of storage volumes that is the destination for backed up or archived files. ADSM supports two types of storage pools: one containing fixed random access media and one containing sequential access media.

An ADSM volume is the basic unit of storage space for a storage pool. An ADSM volume can be an AIX logical volume, a file in a file system, a tape, a cartridge or an optical disk. Each volume is identified by a unique volume identifier.

2.1.2 Storage Management Policies

ADSM allows the management of backup and archive processes based on *policies*. You can decide how granular you want your policies to be. You can establish an overall system policy, policies by department, or by user, or file name. Default policies are predefined in ADSM.

ADSM storage management policies identify information such as:

- When files are eligible for back up or archive
- How many backup versions to keep
- How long to retain backup versions or archive copies
- Where to place backup versions and archive copies in data storage

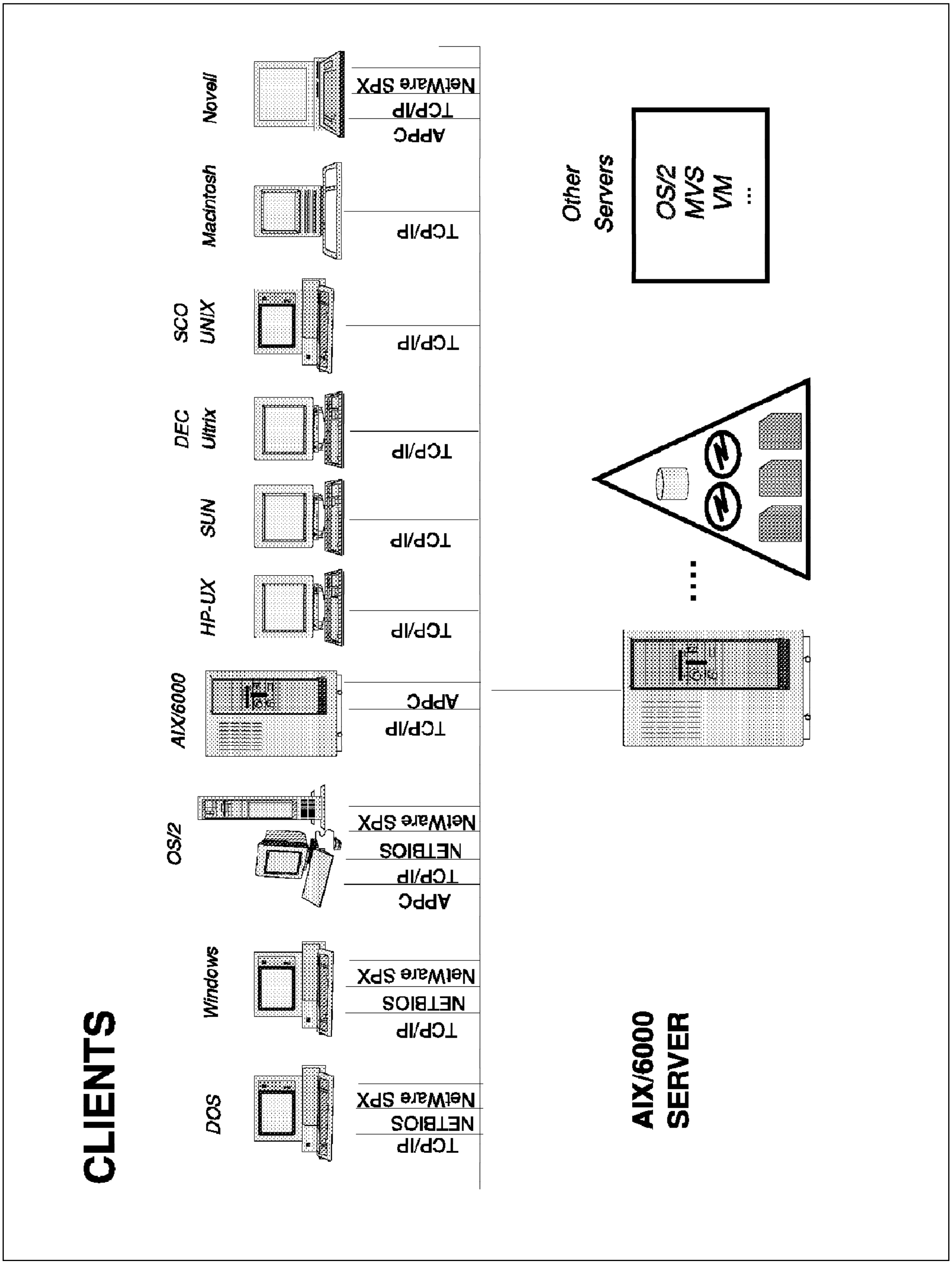


Figure 3. ADSTAR Distributed Manager Server and Client Versions

Policy Domain

A policy domain contains policy sets, management classes, and copy groups that will be used by a particular group of users in your enterprise. A policy domain provides a logical way of managing backup and archive policies for a group of client nodes. It contains one or more policy sets, with only one policy set active at a time.

A policy domain contains the following information:

- The name of the policy domain
- The backup retention grace periods
- The archive retention grace periods

Policy Set

A policy set provides a logical way of grouping management classes. A policy set contains one or more management classes. Only one policy set can be active at a time in a given policy domain. When you define a policy set, you specify:

- The name you want to assign to the policy set
- The description of the policy set

Management Class

A management class is a set of management rules defining backup and archive characteristics. There is a concept of binding the management class to the file when it is backed up or archived. This means that the management class is associated with that file. Users can use the default management class or explicitly select a management class that is within the active policy set to which they have access. You specify one default management class for each policy set.

Copy Group

The copy group provides the ability to specify the management criteria for backup and archive.

A management class can contain two copy groups:

- A backup copy group
- An archive copy group

A copy group controls:

- The destination, where files are to be stored
- Whether the file should be backed up while in use by another process
- The minimal interval between backup or archive operations
- The number of backup versions while the original file still exists
- The number of backup versions after the original file is deleted
- The number of days to maintain all backup versions except for the most recent version
- The number of days to maintain the last remaining backup version of a file no longer on the client′s file system
- The number of days that an archive copy is retained

Figure 4 shows an example of default policy definitions, including the policy domain, the policy set, the management class, and copy groups.

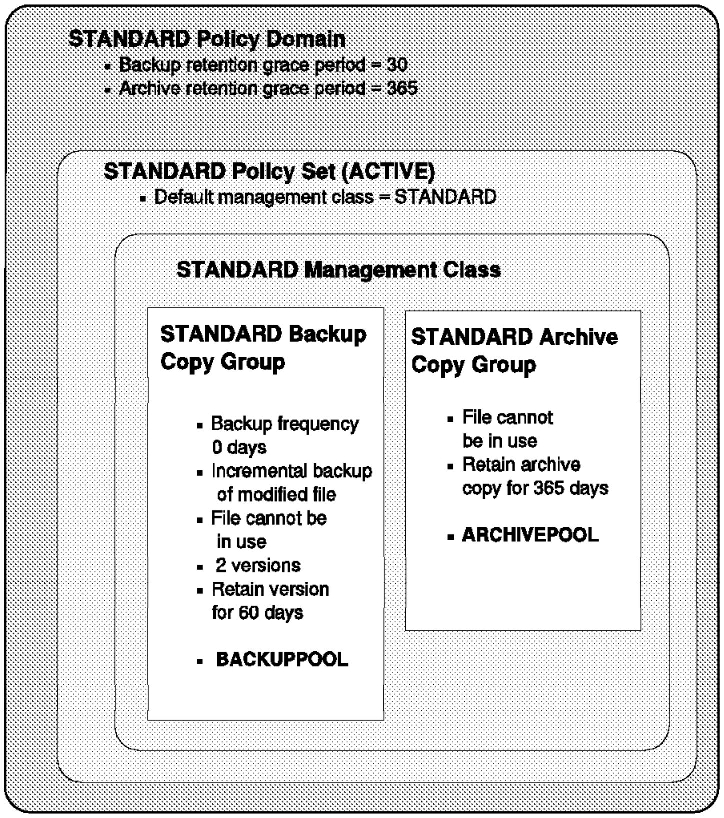

STANDARD Policy Domain
• Backup retention grace period = 30
• Archive retention grace period = 365
STANDARD Policy Set (ACTIVE)
• Default management class = STANDARD
STANDARD Management Class
STANDARD Backup Copy Group
• Backup frequency 0 days
• Incremental backup of modified file
• File cannot be in use
• 2 versions
• Retain version for 60 days
• BACKUPPOOL
STANDARD Archive Copy Group
• File cannot be in use
• Retain archive copy for 365 days
• ARCHIVEPOOL

Figure 4. Storage Management Policy Definition

2.1.3 Central Scheduler

The central scheduler is a function that allows an administrator to schedule backup and archive operations from a central location. The operations can be scheduled on a periodic basis or an explicit date. The administrator coordinates schedule services with the storage management policies.

To define a schedule, an ADSM administrator specifies several parameters. First the administrator specifies the type of service:

- Incremental backup
- Selective backup
- Archive

Then the administrator specifies the following:

- An initial start date for scheduled service
- A startup window for the scheduled service
- How often to run the scheduled service

Each schedule must be associated with one or more client nodes.

A schedule event log is maintained in the server database. Whenever a schedule process starts or fails, an event record is written to the log. An administrator can query the log to determine whether scheduled events completed successfully or not.

Two types of scheduling modes are supported:

- Client polling

  With client polling, a client periodically queries the server for a scheduled operation and the date and time the operation is to start. The server sends this information to the client.

- Server-prompted

  Server-prompted scheduling is supported for all client workstations that use TCP/IP to communicate with the server. With server-prompted scheduling, the ADSM client registers its TCP/IP address with the server and then waits to be prompted by the server to begin the scheduled operation. The client then starts and executes the backup or archive operation.

2.1.4 Nodes

An ADSM node is a unique name for a workstation. In the context of SP2 an ADSM node is usually equivalent to a SP2 node. All ADSM nodes must be registered at the ADSM server before an ADSM client can establish a session with an ADSM server. During registration, nodes are associated with one domain.

2.2 Components of ADSM/6000

The main service provided by ADSM/6000 is to maintain backup versions of your workstation files that you can easily restore if the original files are damaged or lost, and to archive files that you do not currently need on your workstation.

The major components of ADSM/6000 are:

A server program which allows an IBM RISC System/6000 host system to act as the backup and archive server for workstations.

An administrative client program which allows administrators to control and monitor server activities, define storage management policies, and set up schedules to provide backup and archive services at regular intervals.

A backup-archive client program which allows end users to perform backup/restore and archive/retrieve operations.

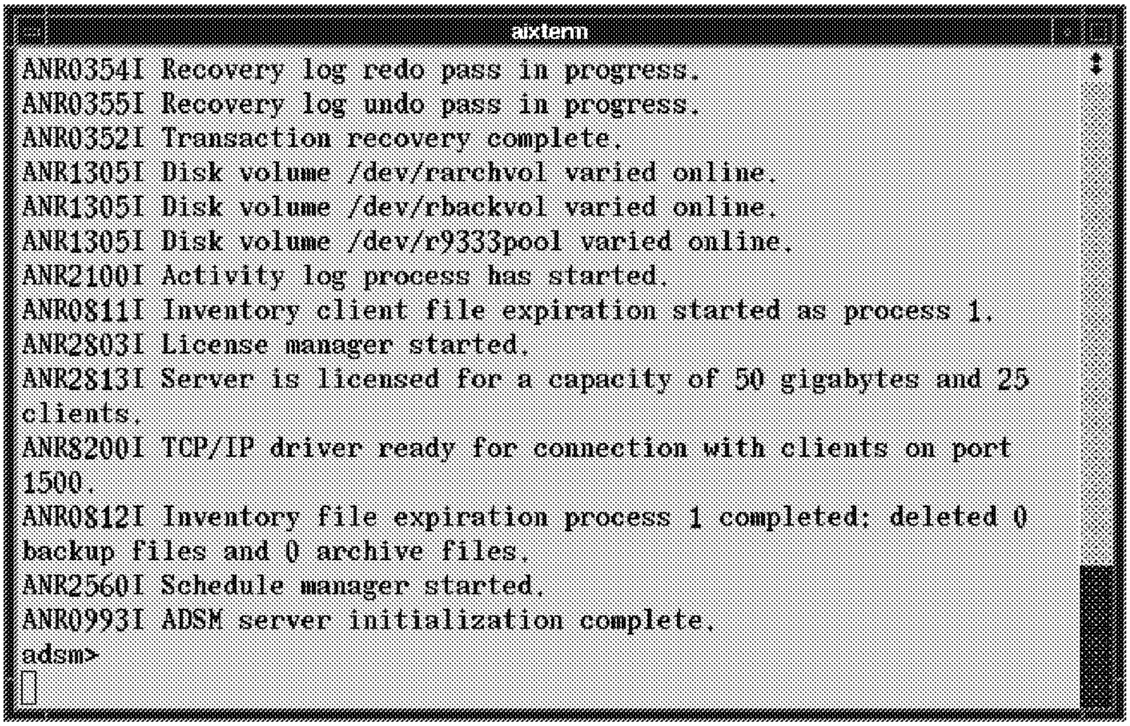
An application programming interface (API) that can be used to enhance existing applications with storage management services.

ADSM provides a command line interface that can be used as an alternative to the graphical user interface to perform ADSM functions.

2.2.1 The ADSM Server

The ADSM server component provides storage resources and services for the backup/archive clients. Users can backup their files onto the storage resources, such as disk, tape or optical devices which are managed and monitored by the ADSM server. Data can be moved automatically through the storage hierarchy onto less expensive media with ADSM′s migration function.

The ADSM administrator can manage the server from a server console, shown in Figure 5 on page 14, or from an administrative client. The server console is the terminal from which an administrator starts the ADSM server in the foreground.



```
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR1305I Disk volume /dev/rarchvol varied online.
ANR1305I Disk volume /dev/rbackvol varied online.
ANR1305I Disk volume /dev/r9333pool varied online.
ANR2100I Activity log process has started.
ANR0811I Inventory client file expiration started as process 1.
ANR2803I License manager started.
ANR2813I Server is licensed for a capacity of 50 gigabytes and 25
clients.
ANR8200I TCP/IP driver ready for connection with clients on port
1500.
ANR0812I Inventory file expiration process 1 completed: deleted 0
backup files and 0 archive files.
ANR2560I Schedule manager started.
ANR0993I ADSM server initialization complete.
adsm>
```

Figure 5. Server Console Interface

The two key components of the ADSM server are:

Storage Pools
Storage pools contain the data of client files that have been backed up or archived. Storage pools are built on disk, tape, or optical devices. The storage pools can be chained in such a way that when a storage pool is getting close to overflow the data migrates to another storage pool.

Database
The database is the heart of the server. It contains:

- Information about registered client nodes
- Policies assigned to client nodes
- Schedules and their association with client nodes
- Event records
- Activity log records
- Information about ADSM volumes
- Data storage inventory

The database has all of the features associated with a database management system. The database is critical to the operation of ADSM/6000, and many features are built-in to ADSM to help maintain the availability, integrity and performance of the database (see also Section 15, “ADSM Availability Issues” on page 127).

- Recovery Log

  Keeps track of all changes made to the database so that if a system failure occurs, a record of the changes would be available in the log.

- Mirroring

Allows the administrator to configure the server so that up to three copies of the database and recovery log are maintained at all times. If a mirrored volume encounters a media failure, the server automatically places the failing volume offline and continues database operations using the other mirrored copies.

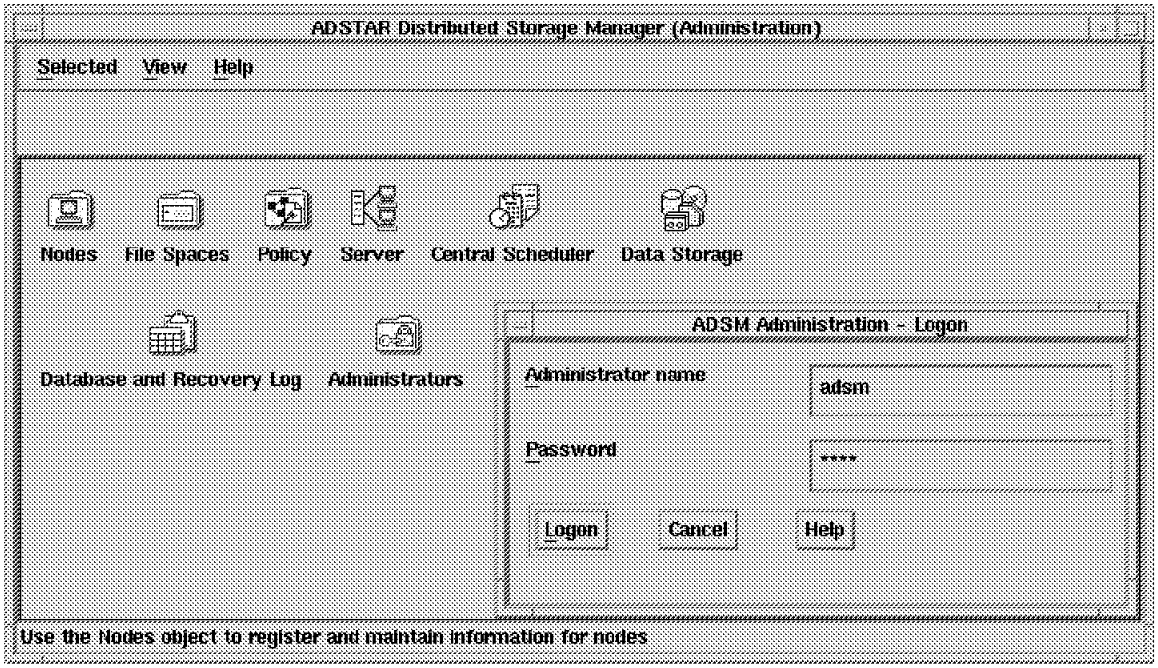
- Export/import

  This function creates a self-describing copy of specified server information. Information that can be exported includes:

  – Administrator information
  – Client nodes definition
  – Policy information
  – Backup and archive data

2.2.2 Administrative Client

An administrative client is a program that allows administrators to control and monitor the server through administrative commands. An administrative client passes commands through an administrative command line or can use a graphical interface, as shown in Figure 6.


ADSTAR Distributed Storage Manager (Administration)
Selected
View
Help
Nodes
File Spaces
Policy
Server
Central Scheduler
Data Storage
Database and Recovery Log
Administrators
ADSM Administration - Logon
Logon
Cancel
Help

Administrator name: adsm
Password: ****

Use the Nodes object to register and maintain information for nodes

Figure 6. Administrative GUI

The administrative client program can be installed on most ADSM clients. However, workstations running AIX/6000 3.2 or later and OS/2 2.0 or later are needed to exploit the graphical interface capabilities of the administrative client. In the context of the IBM POWERparallel Systems SP2, the administrative client should be available from any SP2 node and from the SP2 control workstation.

The administrative client allows an administrator to perform administrative tasks from anywhere within the network.

This program gives the administrator the ability to:

- Define storage management policies for backup-archive

- Define and manage ADSM server storage
- Define and set up the ADSM server database
- Define ADSM backup strategy
- Centrally schedule and monitor backup and archive operations
- Analyze server events

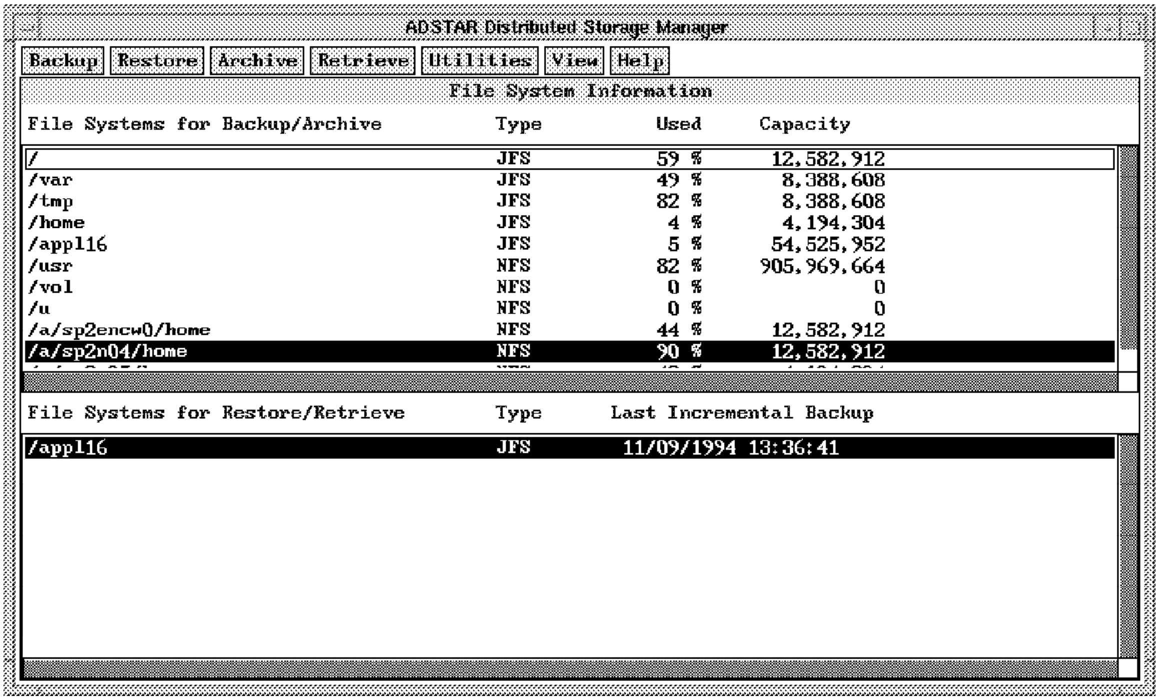
ADSM provides different levels of privileges that can be granted to an administrator:

- *System administrator* with overall authority.
- *Policy administrator* controls the management policies.
- *Storage administrator* controls the storage pools and databases.
- *Operator administrator* controls the operation of the server.
- *Analyst administrator* analyzes server events.

The administrative client communicates with the ADSM server through the same communication protocol as the backup/archive ADSM client.

2.2.3 Backup/Archive ADSM Client

The main functions of ADSM are backup/restore and archive/retrieve. Users can backup/restore and archive/retrieve files using a command line interface or the following GUI interface:



File Systems for Backup/Archive	Type	Used	Capacity
/	JFS	59 %	12,582,912
/var	JFS	49 %	8,388,608
/tmp	JFS	82 %	8,388,608
/home	JFS	4 %	4,194,304
/appl16	JFS	5 %	54,525,952
/usr	NFS	82 %	905,969,664
/vol	NFS	0 %	0
/u	NFS	0 %	0
/a/sp2encw0/home	NFS	44 %	12,582,912
/a/sp2n04/home	NFS	90 %	12,582,912

File Systems for Restore/Retrieve	Type	Last Incremental Backup
/appl16	JFS	11/09/1994 13:36:41

Figure 7. User Interface

In the context of the IBM POWERparallel Systems SP2 the ADSM AIX client program can be installed on any node of the IBM POWERparallel Systems SP2 and on the SP2 control workstation.

2.2.4 Application Programming Interface (API)

The ADSM API is a set of functions that applications running on a client platform can call to store, query and retrieve objects from ADSM server storage. This allows existing applications to backup or archive valuable data in any format specified by the application programmer. The standard ADSM AIX client requires that data be stored in a file in one of the AIX file systems. The ADSM API allows applications, such as databases, to use ADSM services for non-file data and provides a entire new view of usability compared with the standard ADSM client.

You can use the ADSM software developer′s kit to write your own ADSM client. The kit consists of C-language callable services, sample programs, documentation and API code.

2.3 ADSM Main Functions

ADSM/6000 provides the service to backup/restore and archive/retrieve all ordinary files from local and remote file systems, including:

- Journaled File System (JFS)
- Network File System (NFS)
- Andrew File System (AFS)
- Distributed File System (DFS)[1]

ADSM does not back up or archive character special files, block special files, or FIFO files.

2.3.1 Backup and Restore

The backup process creates a copy of a client file on the ADSM server. The backup process also backs up the directory in which the file resides.

You can select:

- *Incremental backup*

  Incremental backup sends to the server the files that have changed since the last backup. The first time an incremental backup is done, all files are sent to the ADSM server. This is a full backup. ADSM determines that a file has changed if any of the following has changed:

  – File size
  – Date and/or time stamp
  – File owner
  – File group
  – File permission
  – Attribute change time

- *Selective backup*

  Selective backup specifies which files a user wants to back up. A selective backup can consist of:

  – Backup by file specification

[1] Files from DFS can be backed up and archived. When restored or retrieved, the access control lists (ACL) information is lost.

  – Backup by directory tree

  Wildcards (*) are allowed in the file specification.

The files are backed up according to policies that the administrator has defined. The policies define, for example, how many backup versions should be retained in the ADSM storage pools, how long to retain those versions and where to back up files.

Restore is the process of copying a backup version from the ADSM server to the ADSM client.

You can specify the files you want to restore by selecting:

- A particular file by name, or a group of files whose name match a specified item.
- One or more directories. You can then assemble a group of files to restore from those directories.
- A particular directory. You can then restore all the files in that directory.

A user can back up any file he owns. The root user can back up all files system-wide.

2.3.2 Archive and Retrieve

Archiving files is similar to backing them up in that both operations store copies of files from the node on ADSM storage.

The key differences between backing up a file and archiving a file are that:

- The user can erase the original file after archiving it. The archived version is retained as long as specified in the management class.
- There is no version attribute in the archive copy group. Archived copies of files are never replaced with more current versions, but are preserved exactly as you store them.

Usually, an archive operation is done less frequently than a back up operation. The ADSM administrator can set up the automatic archive services provided by ADSM.

Retrieving files is the complement of archiving them. When you retrieve a file, ADSM obtains a copy of the archived file and places it on your node.

A user can archive any file to which he has read access, even if he is not the owner of the file. If the user retrieves a file that he has archived, he receives a copy of the file into one of his own directories. So, he becomes the owner of this retrieved copy that has his user-ID as owner′s name. The root user can archive any file system-wide. If the root user retrieves a file, the field identifying the owner of the file is set to the user-ID of the owner at the time the file was archived.

Section 3. SP2 Overview

This section provides a brief overview of the IBM 9076 Scalable POWERparallel System (SP2) and describes its hardware and software.

3.1 Hardware

The IBM 9076 Scalable POWERParallel SP2 System is a distributed memory parallel system composed of processor nodes, which are RISC System/6000 processors. They are housed in frames and are interconnected by one or several communication networks, such as Ethernet, token ring, or FDDI. Also, they are optionally interconnected by the high-performance switch (HPS), which provides a fast communication path between SP2 nodes.

The SP2 is managed from a control workstation that is an IBM RISC System/6000 running the Parallel System Support Programs (PSSP), which includes software functions to monitor the SP2 from a single point of control. Figure 8 on page 20 shows a simple one-frame system with the following connections:

- The high-performance switch
- The external LAN, for instance an Ethernet LAN, that interconnects the SP2 nodes and the control workstation
- The RS-232 line that connects the control workstation to the frame supervisor card.

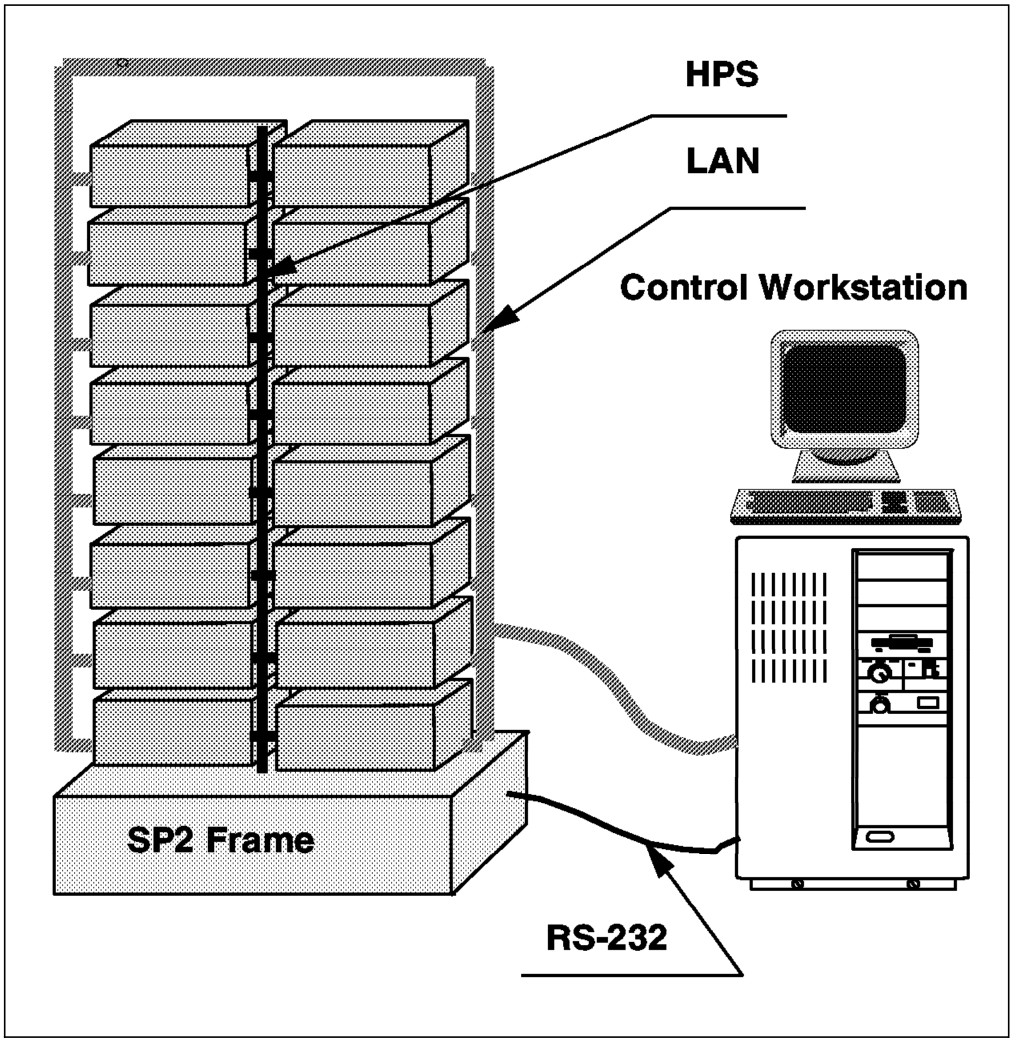


Figure 8. 9076 SP2 Network Configuration

3.1.1 Nodes

The nodes are available in two different types: thin nodes and wide nodes. They differ in computing power, maximum memory and I/O growth capability.

Thin Nodes
The thin node is equivalent to a RISC System/6000 Model 390, containing a POWER2 processor operating at 66.7 MHz with 64KB of data cache and 32KB of instruction cache. The 9076 SP2 also supports the SP1 nodes, equivalent to a RISC System/6000 Model 370, containing a POWER processor operating at 62.5 MHz. The thin node has four Micro Channel adapter slots and can have up to two SCSI disks packaged internally. The Ethernet adapter is integrated on the system board. Each thin node is half the width of a wide node. Each frame can contain up to 16 thin nodes, arranged in two columns of up to eight nodes each.

Wide nodes
The wide node is equivalent to a RISC System/6000 Model 590, containing a POWER2 processor operating at 66.7 MHz with 256KB of data cache and 32KB of instruction cache. The wide node has eight Micro Channel Adapter slots and can have up to four SCSI disks packaged internally. The Ethernet

adapter takes up one slot. Each wide node is twice the width of a thin node. Each frame can contain up to eight wide nodes.

The system can contain any number of wide nodes and an even number of thin nodes (thin nodes are always sold in pairs).

3.1.2 High-Performance Switch (HPS)

The HPS is an optional network connecting the nodes. It consists of a switchboard per frame, housed at the bottom of the frame. For SP2 configurations with up to 80 nodes, high-performance switch boards are directly interconnected. Larger configurations require an external growth cabinet housing intermediate switchboards. The nodes in a frame are connected to the switchboard through a Micro Channel Adapter. The switchboards are connected to each other through external cabling.

On IBM 9076 SP2 systems, the high-performance switch is connected to nodes through the HPS Adapter-2, which allow high-speed communications between processes running on separate nodes of the configuration. System functions and application processes can communicate through the high-performance switch using the internet protocol (IP) in kernel mode. PSSP includes another protocol used by parallel programs developed with IBM AIX Parallel Environment or IBM PVMe. Then, the processes of these parallel programs communicate through the switch in user mode for better performance.

If two processes on different nodes, which belong to a parallel application, communicate through the HPS in user mode, then all other processes on these nodes can use the HPS only in kernel mode (IP). See 3.2, “SP2 Software” on page 22 for more information on this subject.

3.1.3 Control Workstation

The control workstation is a RISC System/6000 workstation with a graphic monitor. It runs the PSSP software and provides a single point of control for system management and monitoring. It is connected to the nodes through a LAN, and to the frame supervisor card through a RS-232 serial line.

3.1.4 SP2 Ethernet

On 9076 SP2, nodes are connected to the control workstation through an Ethernet LAN named the SP2 Ethernet. This LAN is used to install nodes, manage the system and provide a dataless node with the operating system through NFS. It can be attached to other site networks. However, a better approach is to connect SP2 nodes to site networks through another network: for instance, Ethernet, token ring, and FDDI can be used on purpose.

3.1.5 Serial Lines

In an SP2 frame, the nodes and the high-performance switch include supervisor cards, which are connected through an internal bus to the frame supervisor card.

Each frame supervisor card is connected through a RS-232 line to a serial port of the control workstation. A multiport adapter can be required on the control workstation to support multiframe configurations. The serial connection is used by the PSSP system monitor component, which provides hardware control functions through the RS-232 lines. So, the administrator can access the hardware information, and can send hardware commands from the control workstation.

3.2 SP2 Software

Each node of SP2 runs a full image of the AIX/6000 Version 3.2.5 operating system. As such, it can run virtually any AIX application. The Parallel System Support Programs (PSSP) software is installed on the control workstation and allows the administrator to manage and monitor the SP2 system. It includes several components, such as:

System Data Repository (SDR)

The SDR is a common data repository that provides storage and retrieval from the control workstation and all the nodes for system data. System and node configuration data, job data, and other system data is stored in the SDR. Command line and C library interfaces are provided. SDR uses the object data manager (ODM). ODM is the database manager of AIX. Most access for SDR data is through the centralized management interface (CMI).

System Management (SMAN)

PSSP provides a set of tools that enhance the traditional AIX management tools. It includes:

Network Time Protocol

NTP is a network protocol designed to keep large computer networks in synch with a single time source. It runs on Internet but can be used even if you are disconnected from Internet and want to use your own time system. It is public domain software.

File Collections

Using file collections, groups of files can be defined as a collection from a single point, and any changes to any files in that collection will be automatically propagated to all the systems that have that collection installed. The file collection technology is based on *sup* (the Software Update Protocol from Carnegie Mellon University) and uses a Perl front end named *supper*. File collection servers are arranged in a tree structure for scaling. The SP2 system comes with predefined collections for user administration files, such as /etc/passwd, and for root control files, such as /etc/services. The administrator can add any files to these collections. Additional collections can be defined as well.

User management support

Users can be managed in two ways:

- The Network Information System, also named Yellow Pages: it is a client/server mechanism used to centralize user definitions. A tree is defined rooted at a master server. The master server distributes the user′s definitions (stored in files named *maps*) to slave servers. All clients access a server (either master or slave) to check the user′s definition at login time.
- The file collection mechanism can be used to manage /etc/passwd, automatically distributing it to all the nodes.

Auto Mount Daemon (*amd*)

Amd is a mechanism to dynamically mount user′s home directories when users log in. It can mount either from a local disk or from a remote NFS server. It is often used in NFS environments together with the Network Information System. It sets up symbolic links from /u/* to either /home/* or /a/*NFSservername*/home/* respectively for local and remote mounts.

Print support

SP software provides some simple scripts that optionally replace the AIX print commands and use *rsh* to shift printing to a print server.

Login Control

New commands are provided with the SP2 to allow the administrator to prevent a user or group of users from logging onto a node or group of nodes.

Accounting

The SP2 support for accounting builds upon standard AIX accounting and provides two main capabilities:

- Accounting record consolidation
- Parallel job accounting

Partial reduction of accounting data is done on each node before the data is consolidated on an accounting master. Nodes are configured into groups called classes. All the data from a given class is consolidated together. The command *nrunacct* has been developed for that purpose and is typically scheduled to run every night. SP2 also provides an optional mechanism to charge for nodes that have been assigned for exclusive use, for example jobs running the user space communication protocol. If the support is enabled, an accounting record for a marker process is created before and after the job is run.

System Monitor (SMON)

The System Monitor is a graphic application that runs on the control workstation and allows the system administrator to monitor the hardware and the software running on each node and to perform many system operations like:

- Power on/off any node
- Shutdown/reboot any node
- Connect to the node console
- Display node′s LEDs

The System Monitor keeps its information in the SDR.

Distributed Shell (dsh)

The distributed shell (*dsh*) is a new command provided to execute commands on all the nodes or a subset of nodes. It is based on *rsh* and, hence, requires *rsh* privileges. *dsh* allows the user to execute all the commands in parallel but will keep output from each node together. By selecting a fanout of one, the user can also get sequential execution. As a default, *dsh* will only operate on the nodes that respond to the System Monitor software running on the control workstation.

Centralized Management Interface (CMI)

The centralized management interface (CMI) is a SMIT based interface for managing and operating the SP2 system. The CMI provides functions to:

- Enter, change and show system and site configuration data
- Configure node installation parameters
- Configure boot/install servers or /usr servers
- Configure the control workstation
- Manage the high-performance switch
- List node hardware data
- List node network data or node file system data
- Manage users

Communication Subsystem Support (CSS)

The CSS is a set of software layers that support the communications through the high-performance switch. It includes several features, such as:

- HPS adapter device drivers
- Software tools to initialize the HPS, detect failing nodes, and detect failing links
- Interfaces with communication protocols that can be used to communicate through the high-performance switch. Two protocols are supported by CSS:
  - The internet protocol (IP)

    On SP2 systems, an application that communicates through a network using TCP/IP interfaces can execute message passing operations either through an external network or through the high-performance switch. When the chosen communication path is the high-performance switch, the message exchange operations using the IP protocol are oriented to the HPS Adapter-2 device driver, which enwraps IP messages with the high-performance switch protocol. So, the applications using the IP protocol can transparently use the high-performance switch to achieve high-speed communications and data transfer.
  - The *user mode* protocol

    This protocol is used in message passing library subroutines provided with IBM AIX Parallel Environment and IBM PVMe to develop and run high-performance distributed parallel programs. Communications are directly executed from the user space without any system call. Therefore, TCP/IP layers are bypassed, and imply greater bandwidth and lower latency.

3.2.1 SP2 Software Environment

The SP2 can be setup in two different software environments. The first one is to operate the SP2 as a cluster of IBM RISC System/6000 nodes working in a client/server environment. In this mode of operation, a single file subtree does not exist, and the SP2 software is only used for hardware control and administrative purposes. The nodes are operated as independent workstations. Nodes that have access to large physical storage devices will be configured as ADSM servers.

The second environment is to setup the SP2 as a cluster of IBM RISC System/6000 nodes, and the nodes will be managed as a single image system. Through the AIX Parallel System Support Programs (PSSP) for SP2, the nodes are managed as follows:

Single user domain

The SP2 cluster is administered as a single user domain. Users are added from the control workstation once. Through either NIS or the file collection mechanism, the user account information and password is propagated to other nodes of the SP2. Users are not associated with any particular node. Instead, users are able to log on from any node of the SP2. The users see exactly the same user identity *(uid)* and remain as the member of the same groups, that is, same group identity *(gid)* as though they logged on to one system.

Globally accessible data

Most of the user′s data is globally available at identical points in the file naming tree. This is accomplished by using NFS and the automount utility (*amd*) of the SP2 system. For example, we have defined some users in the */etc/amd/amd-maps/amd.u* like this :

Sample of amd.u map

```
/defaults type:=link sublink:=${key}
spuser1 host==sp2cw0;type:=link;fs=/home  \
        host!=sp2cw0;type:=nfs;rhost:=sp2cw0;rfs=/home
spuser2 host==sp2cw0;type:=link;fs=/home  \
        host!=sp2cw0;type:=nfs;rhost:=sp2cw0;rfs=/home
```

This means if user *spuser1* logs on to node sp2cw0, his /home/spuser1 directory will be symbolically linked to /u/spuser1. If he logs on from nodes other than sp2cw0, the /home directory will be auto-mounted from sp2cw0 through NFS.

From the user′s point of view, the user′s home directory is always identical on all nodes of the system and is globally available from any node.

Loadlevelling

User′s work units can be assigned to different nodes at different times based on the loading of the system. For example, with Loadleveler, a user may have his job executed on node A today but on node B the next day. The allocation of nodes and the execution of work on the nodes is transparent to the users.

It is also common to use AIX shared /usr file server in the SP2 environment to reduce disk space consumption and make system management easier.

Section 4. ADSM on SP2

ADSTAR Distributed Storage Manager is the IBM solution for storage management in distributed environments with server and client versions available on many platforms. Figure 3 on page 10 summarizes the current server and client versions of ADSM that are already available.

4.1 Roles of ADSM/6000

Three major roles are possible for the ADSM/6000 on the SP2:

ADSM server only
The ADSM server runs on an SP2 node that has access to storage devices, such as tape, disk or library. Supported workstations on a network are connected through a gateway node on the SP2 to the ADSM server. The workstations in this case are external ADSM clients, and the ADSM server is within the SP2.

ADSM clients only
The ADSM clients could be any node within the SP2 system. The ADSM server could be an IBM host system (MVS, VM or VSE) or an RS/6000 server external to the SP2 cluster. The ADSM server is accessed through a gateway node within the SP2.

ADSM server and client
In this scenario, both ADSM client and server are configured within the SP2. The data to be managed resides on the file server nodes and on the local disks of some of the nodes of SP2. These nodes are the ADSM clients. One or more ADSM servers can be configured on SP2 nodes that have access to the physical storage devices, such as disks subsystems, tape subsystems and optical disk subsystems. In this book, we mainly focus on this environment.

4.2 Data on the IBM POWERparallel Systems SP2

It is important to understand where the data resides in the SP2 to plan for an efficient backup strategy. Those SP2 nodes that carry data should be the first choice for installing an ADSM client. Of course, data on any node in the SP2 can always be mounted with NFS, but for performance reasons it is not advisable to move the data from an SP2 data server first to an ADSM client and then to an ADSM server. It is better to install the ADSM client directly on the SP2 node that contains the data in its local file system.

Data normally resides on one of the following nodes within the SP2.

Data server nodes
These nodes have physical access to large amounts of storage, such as IBM 9333 disk subsystems. The server may contain user or application data used for the whole system. The data is accessible by applications through Network File Systems (NFS), Andrew File System (AFS) or Distributed File System (DFS).

/usr server node
/usr servers are a special type of file server that exports the /usr file system to other SP2 nodes where local disk storage is limited. By using the shared /usr file system, the administrative effort required to maintain the SP2

software environment is greatly reduced. This approach also results in a saving of local disk space on the client nodes.

Nodes with local data

These nodes have local data residing on their internal hard drives to serve applications. The data on these nodes is not exported to other nodes but is used by applications that run on the same SP2 node, and that distinguishes them from the data server nodes. This data has to be backed up properly within the SP2. Figure 9 shows an example of how data can be spread accross different nodes of the SP2.

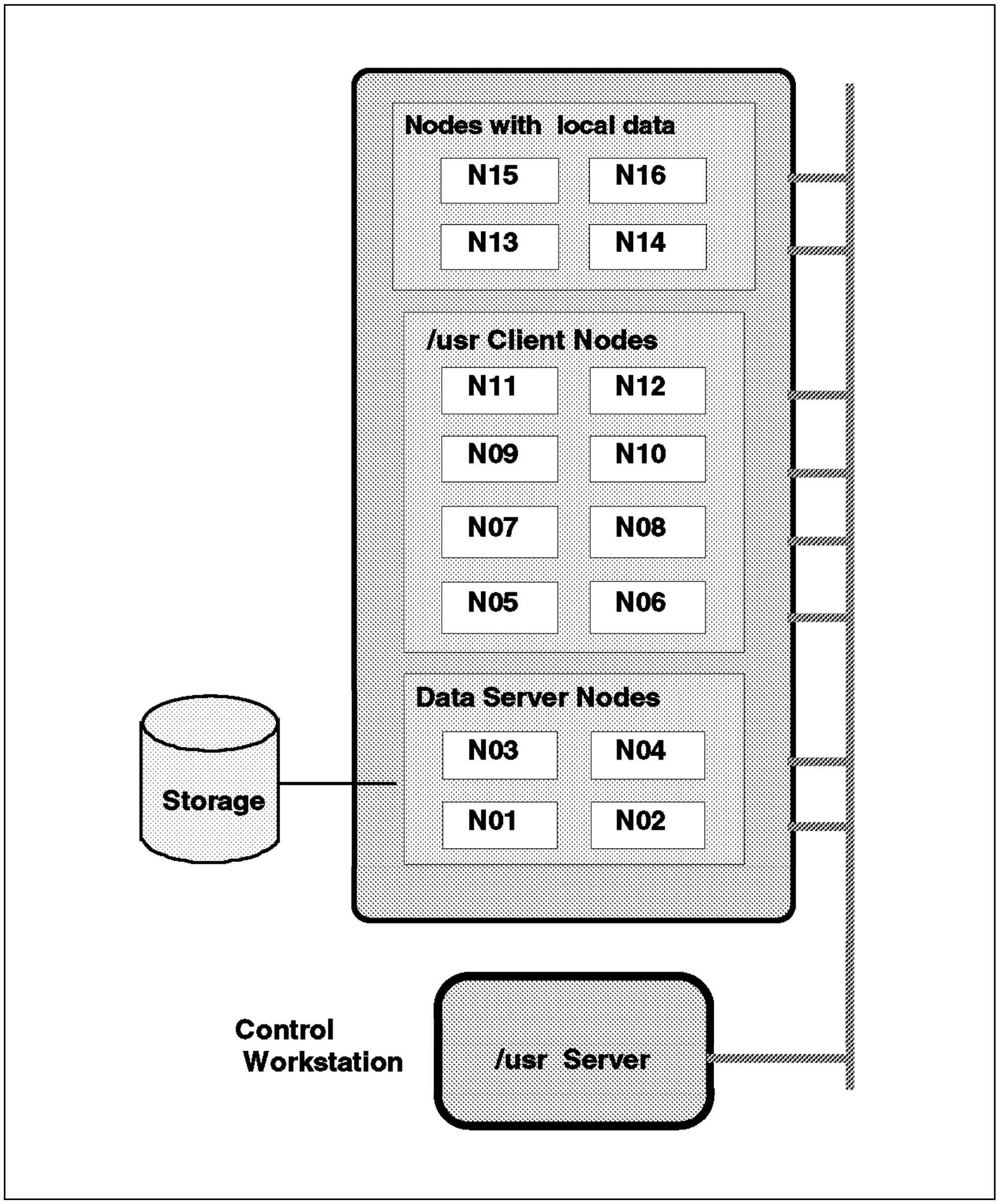


Figure 9. Example of SP2 with Data Spread on Different Nodes

4.3 Selecting Storage Devices

Installation of ADSM/6000 on the IBM POWERparallel Systems SP2 requires some careful planning when selecting I/O hardware. Check which I/O hardware is available to ADSM server nodes. Appendix D, “Device Support” on page 157 provides an overview of the storage devices currently supported by ADSM/6000. It may be required to install several ADSM servers to get the throughput expected from the backup system.

Selection of the components to make up the storage subsystem is not easy and often is a trade-off. First, the application requirements need to be considered in terms of their storage necessities; the mix of storage device types then needs to be decided; and finally specific products need to be chosen.[2]

Selecting the correct products for inclusion into a storage management subsystem involves consideration of several points:

- Cost per megabyte

  The cost per megabyte of storing information on a device is always important. This cost is usually proportional to the speed of access to data on the device, and generally also proportional to the capacity. If rapid access to data is essential, then the cost will be higher.

- Access frequency

  This figure represents the number of times data on the device will be accessed in a given period. The higher this figure, the higher the data rate, access times, and reliability usually need to be. Devices supporting frequent access are usually required for interactive types of applications.

- Access density

  This value describes the number of I/O requests per gigabyte of data. This value needs to be considered in conjunction with the access frequency. High access density and low frequency suggests using optical storage; high density and high frequency suggests disk; low density and low frequency suggests tape; low density and high frequency may involve other decision points, such as cost.

- Access type

  This describes the required method for access to the data on the device, and can be sequential or random. Random access means accessing small amounts of data at many points on the device, while sequential access means accessing large amounts of data from few points. The difference is mainly with just how easy it is to locate specific data elements on the device. If it takes a long time to search for the required information, random access is not recommended. Database applications tend to involve random requests, while backups or restores are good examples of sequential operations.

- Data rate

  This describes how rapidly the data can be obtained from the device (once located). Interactive applications tend to require a high data rate, while batch applications can usually tolerate lower data rates.

[2] Help for this step is available from a redbook called *AIX Storage Management*.

- Online life

  This describes how long data will need to be accessible on the device and is sometimes known as data age. Data age is usually inversely proportional to access frequency, and therefore high values here usually imply the use of low cost, low speed, high capacity devices.

- Interchange requirements

  Will the information stored on this device be exchanged with other systems? If this is the case, then some form of removable media will be required.

- Longevity

  If archive, backup or even reference information is to be stored for long periods of time, then the integrity of the media used is important. Most media have a defined shelf life, after which degradation of stored information is likely.

- Reliability

  This is an extremely important consideration, particularly when mission critical information is being considered.

Chapter 2. ADSM Installation and Basic Customization

This chapter describes ADSM/6000 server and client installation on the IBM POWERparallel Systems SP2. It explains ADSM basic customization and suggests ways to verify the configuration before going on with advanced customization.

We discuss the installation of the ADSM server and client code:

- On full-AIX

  This means we install the server code on the local disk of the ADSM server node, and we install the ADSM client code on SP2 nodes.

- On a /usr shared directory

  This means that ADSM server and client codes are installed once on a /usr data server, and each ADSM client or server accesses the code through a /usr shared directory on the data server.

We describe the basic customization on the ADSM server and client and then explain how to verify the customization and execute a simple backup.

Section 5. ADSM Customization Steps

This section is a general overview of the steps that you must follow when installing and customizing ADSM/6000. The purpose of this section is to give you a general checklist. Detailed information about the installation and customization is provided in Section 6, "ADSM/6000 Installation" on page 35.

Additional information about the installation and customization process of ADSM/6000 server and client can be found in:

- *ADSM Users Guide and Reference for UNIX*
- *ADSM/6000 Installing the Server and Administrative Client*
- *Getting Started with ADSM/6000*
- *Getting Started with ADSM AIX Clients*

5.1 Steps for the ADSM Server Customization

The following checklist is an overview of the steps to perform the installation and customization of the ADSM/6000 server on one SP2 node.

1. Install all prerequired software.
2. Install ADSM/6000 code on the server.
3. Set up and customize the ADSM server option file (dsmserv.opt).
4. Load the ADSM/6000 kernel extension.
5. Create the database and log files.
6. Create backup and archive storage volumes.
7. Start the server.
8. Create an ADSM administrator with system authority.
9. Configure all devices used with ADSM/6000 that have ADSM specific device drivers.
10. Define devices in ADSM/6000 (device-class, drives, libraries).
11. Label your tape volumes.
12. Create storage pools and associated volumes.
13. Ask your support center for the iFOR/LS (formerly NetLS) key.
14. Register your license.
15. Register nodes.

There is a lot more that can be customized. However, these steps should be sufficient to get the ADSM/6000 server running.

5.2 Step for the ADSM Client Customization

Here are the steps to install and customize the ADSM/6000 client code:

1. Install prerequired software for ADSM AIX client.
2. Install ADSM AIX client code on SP2 nodes.
3. Customize client files dsm.opt and dsm.sys for each SP2 node.

Section 6. ADSM/6000 Installation

This section describes:

- ADSM/6000 server installation
- ADSM/6000 client installation

and the basic definition of the options files.

In the SP2 configuration, we have considered two types of ADSM server and client installations:

- Installing on a full-AIX system
- Installing on a shared /usr file system

Both of these installation procedures are explained in the following section.

6.1 ADSM/6000 Server Installation

In the ITSO Open Systems Laboratory SP2 configuration we installed the ADSM/6000 server on node 2 and on the SP2 control workstation.

- Node 2 is a full-AIX node with a local /usr file system. The IBM 9333 DASD and the IBM 3490E Model E11 are connected to node 2. This is considered an installation of the ADSM/6000 server on a full AIX system and is described in 6.1.1, “Installation on a Full-AIX Node.”
- The SP2 control workstation is a /usr server for all nodes of the SP2 except node 2. This installation is considered an installation on a /usr server for /usr clients and is described in 6.1.2, “Installation for /usr Clients” on page 36.

6.1.1 Installation on a Full-AIX Node

This procedure describes how to install the ADSM/6000 server code on a full AIX system (all file systems are local). We have tried to emphasize those steps that are specific to the SP2. If you need more general guidelines about installing ADSM/6000 server code, refer to *Getting Started with ADSM/6000*.

6.1.1.1 Installation Procedure

Use the following procedure to install the ADSM server on your node:

1. Log on to your ADSM server node as the root user.
2. Enter SMIT with the following command:

```
SMIT install
```

3. On the first panel, select **Install and Update Software**.
4. On the next panel, select **Install/Update Selectable Software** .
5. On the next panel, select **Install Software Products at Latest Available Level**.
6. On the next panel, enter the name of the tape drive (for example /dev/rmt0) and then press Enter.
7. Select the ADSM code and press Enter.

If you don′t have a tape connected to your ADSM server, copy the install images from the tape drive to the /usr/sys/inst.images directory on the SP2 control workstation. Then export the /usr/sys/inst/images directory from the SP2 control workstation and mount the file system from your ADSM/6000 server target node. Use the following command to mount, where sp2cw0 is the SP2 control workstation hostname.

```
        mount -n sp2cw0 /usr/sys/inst.images /mnt
```

You can now execute SMIT on the ADSM/6000 server node. Invoke the SMIT installation submenu and specify /mnt as the input device.

6.1.1.2 ADSM Server Options File

After installing the ADSM/6000 server, you must create the server options file, *dsmserv.opt*. There is a sample in */usr/lpp/adsmserv/bin/dsmserv.opt.smp*, and this file should be copied to */usr/lpp/adsmserv/bin/dsmserv.opt*. The options file allows you to define the communication method used between the ADSM server and client and various other parameters. TCP/IP is the default for the communication method, and if you use it, there is no need to customize the dsmserv.opt file for the moment. Copy *dsmserv.opt.smp* file to *dsmserv.opt* file.

```
cd /usr/lpp/adsmserv/bin
cp ./dsmserv.opt.smp  ./dsmserv.opt
```

6.1.2 Installation for /usr Clients

The /usr file system typically contains executable programs. By concentrating the executables on a /usr server system, considerable disk space can be saved on the /usr client systems. It also eases the maintenance of the components of the AIX base operating system (AIX BOS), which are contained in the /usr file system. If the ADSM/6000 server code is installed in the shared /usr file system, the code is executable from all nodes that share the /usr file system. We will not describe how to set up a /usr server on the SP2. This information can be found in SP2 9076 Scalable POWERparallel System Administration Guide.

6.1.2.1 Installation Procedure

The ADSM server installation on a /usr server is not really different from the installation in a full-AIX System. Log on to the node where the shared /usr file system is located, and install the ADSM/6000 server code, as explained in the previous section.

The major difference will be the management and the location of the ADSM/6000 server configuration file (*dsmserv.opt*). The best way to manage individual option files is to create a local directory on each node and put the ADSM/6000 server configuration files into this local directory.

6.1.2.2 ADSM Server Options File

For the ADSM/6000 server options file, we recommend creating a directory called */var/adsmserv* on each node where you want to run the ADSM server program. This directory will contain unique configuration programs for that node. If you have to customize only one ADSM Server, use this procedure:

1. Log on to your SP2 control workstation as root user. Then:

- Create the /var/adsmserv directory on the SP2 control workstation
- Copy /usr/lpp/adsmserv/bin/dsmserv.opt.smp to /var/adsmserv/dsmserv.opt
- Create a symbolic link between the initial option file directory /usr/lpp/adsmserv/bin/dsmserv.opt and the new one /var/adsmserv/dsmserv.opt.

```
mkdir /var/adsmserv
chmod 666 /var/adsmserv
cp /usr/lpp/adsmser/bin/dsmserv.opt.smp /var/adsmserv/dsmserv.opt
ln -s /var/adsmserv/dsmserv.opt /usr/lpp/adsmserv/bin/dsmserv.opt
```

2. Log on as root to the node, where you want the ADSM server code to execute. Create the /var/adsmserv directory and copy the file dsmserv.opt from the SP2 control workstation.

```
mkdir /var/adsm
chmod 666 /var/adsmserv
rcp sp2cw0:/var/adsmserv/dsmserv.opt /var/adsmserv/dsmserv.opt
```

If you have to customize several ADSM Servers, you can create the /var/adsmserv directory and copy the option file on all ADSM/6000 server nodes from the SP2 control workstation using the dsh command.

1. Log on to the SP2 control workstation as root user. Then:

   - Create the /var/adsmserv directory on the SP2 control workstation
   - Copy /usr/lpp/adsmserv/bin/dsmserv.opt.smp to /var/adsmserv/dsmserv.opt and
   - Create a symbolic link between the initial option file directory /usr/lpp/adsmserv/bin/dsmserv.opt and the new one /var/adsmserv/dsmserv.opt.

```
mkdir /var/adsmserv
chmod 666 /var/adsmserv
cp /usr/lpp/adsmser/bin/dsmserv.opt.smp /var/adsmserv/dsmserv.opt
ln -s /var/adsmserv/dsmserv.opt /usr/lpp/adsmserv/bin/dsmserv.opt
```

2. Create a file (for example server.list) on the SP2 control workstation that contains the list of all nodes where the ADSM server should run, such as:

```
sp2sw03
sp2sw04
sp2sw05
```

   This file will be used with the dsh command to specify on which nodes the command should be executed.

3. Now execute the dsh command, create the /var/adsmserv directory on each node and make sure you have write permission on /var/adsmserv. Then copy the dsmserv.opt from the SP2 control workstation to /var/adsmserv, such as:

```
export WCOLL=/server.lists
dsh mkdir /var/adsmserv
dsh chmod 666 /var/adsmserv
dsh rcp sp2cw0:/var/adsmserv/dsmserv.opt /var/adsmserv/dsmserv.opt
```

The ADSM/6000 server is now installed on the /usr server system for use by the /usr client systems.

6.2 ADSM/6000 Client Installation

For the same reasons as those described in the ADSM server installation, you can chose between an installation of the ADSM/6000 client code either on a full-AIX node or on a /usr server.

6.2.1 Installation on Full-AIX Node

If some of your SP2 nodes have their own /usr, you will have to install ADSM/6000 client code locally on each node. So, you will have to run the installp program on each node. The best way is to execute the command from a single point of control using the dsh command. The dsh command will execute commands on any node in your network to which you can issue a command through rsh.

6.2.1.1 Installation Procedure

1. Copy the ADSM/6000 client code image to the directory /usr/sys/inst.images on the SP2 control workstation.

2. Define the hostnames list for use with *dsh*, such as:

```
sp2sw10
sp2sw11
sp2sw12
sp2sw13
sp2sw15
sp2sw14
sp2sw16
```

3. Mount the file system containing your ADSM installation image from the SP2 control workstation to the target nodes:

```
export WCOLL=./client.list
dsh mount -n sp2cw0 /usr/sys/inst.images /mnt
```

4. Execute the installation command using the dsh command.

```
export WCOLL=./client.list
dsh /usr/lib/instl/sm_inst installp_cmd -T iems -L
 -q -a -d /mnt -S ′1.2.0.3  adsm.obj′ -g -X
```

If you want to install the ADSM client code on every node, you don′t have to define a list. Just use the *-a* parameter with the dsh command.

6.2.1.2 ADSM Client Options Files

After installing the ADSM/6000 client code, set up the ADSM/6000 client options files, *dsm.opt* and *dsm.sys*. Those files usually reside in /usr/lpp/adsm/bin. If you want to put all the options files in another directory, link this new directory with the original one.

dsm.sys: ADSM/6000 provides a sample system option file named *dsm.sys.smp*.

- Copy the file to create a default client system options file.

```
cp /usr/lpp/adsm/bin/dsm.sys.smp /usr/lpp/adsm/bin/dsm.sys
```

- Edit the dsm.sys file and define the communication parameters such as:

```
Servername        sp2n02
Commmethod        TCPip
TCPport           1500
TCPServeraddress  sp2sw02.itsc.pok.ibm.com
```

The servername, the communication method, the TCPport, and the TCPServeraddress are required to be defined in the system option file.

dsm.opt: ADSM/6000 provides a sample user option file named *dsm.opt.smp*.

- Copy the file to create a default client user options file:

```
cp  /usr/lpp/adsm/bin/dsm.opt.smp /usr/lpp/adsm/bin/dsm.opt
```

There is no need to edit the *dsm.opt* file for the moment.

6.2.2 Installation for /usr Clients

The installation of the ADSM/6000 client code on a /usr server system is similar to the installation in a full-AIX system. The difference is the handling of the customization files.

6.2.2.1 Installation Procedure

Log on to the node where the shared /usr file system is located and install the ADSM/6000 client code. The procedure is described in 6.2.1.1, “Installation Procedure” on page 38.

6.2.2.2 Options Files

For ADSM/6000 client options files, create a directory called */var/adsm* on each SP2 node where you want the ADSM/6000 client program executed. This directory will contain the node unique configuration files. With this local directory containing ADSM options files, each node can define its own ADSM options.

You can create the directory and copy the option file on all your ADSM client nodes from the SP2 control workstation, using the dsh command.

1. Log on to your SP2 control workstation as root user. Then:
   - Create the /var/adsm directory on the SP2 control workstation
   - Copy the sample option files
   - Create a symbolic link between the initial option file /usr/lpp/adsm/bin/dsm.opt and the new one /var/adsm/dsm.opt

```
mkdir /var/adsm
chmod 666 /var/adsm
cp /usr/lpp/adsm/bin/dsm.opt.smp /var/adsm/dsm.opt
cp /usr/lpp/adsm/bin/dsm.sys.smp /var/adsm/sys.opt
ln -s /var/adsm/dsm.sys /usr/lpp/adsm/bin/dsm.sys
ln -s /var/adsm/dsm.opt /usr/lpp/adsm/bin/dsm.opt
```

2. Edit the *dsm.sys* file and define the TCP/IP communication method, the servername, TCPport address, and the TCPServeraddress, such as in:

```
Servername        sp2n02
Commmethod        TCPip
TCPport           1500
TCPServeraddress  sp2sw02.itsc.pok.ibm.com
```

3. Create a file on the SP2 control workstation that contains the list of your ADSM client node hostnames. This file will be used by the dsh command to specify on which node to execute the command.

4. By using the dsh command, create the /var/adsm directory on each node and provide the correct permission and copy the options files.

```
export WCOLL=/client.lists
dsh mkdir /var/adsm
dsh rcp sp2cw0:/var/adsm/dsm.sys /var/adsm/dsm.sys
dsh rcp sp2cw0:/var/adsm/dsm.opt /var/adsm/dsm.opt
```

The ADSM/6000 client code is now installed on the selected SP2 nodes.

Section 7. ADSM Basic Customization

This section describes the basic customization before ADSM can be used for any operations.

After the code installation you must:

- Create the ADSM database and log.
- Define the storage pool environment.
  - Define devices and libraries.
  - Define storage pools.
- Execute basic administrative tasks.

7.1 ADSM Database and Log

The ADSM database holds information about all definitions. This includes mainly information about the files that reside in the storage pools, but also includes information about the registered client nodes and policies and control information for administrative clients.

The server uses the recovery log to keep a record of all changes made to the database.

After code installation, you must prepare space for the ADSM/6000 database and recovery log. To do this, you can choose between:

- Preparing AIX logical volumes to be used
- Preparing files in an AIX file system to be used

In this redbook, we have chosen logical volumes to be used as space for the ADSM/6000 database and the recovery log. If you need information about how to prepare files for use as ADSM database volumes, refer to *ADSM/6000 Installing the Server and Administrative Client*.

7.1.1 Loading the Kernel Extension

Before running the ADSM/6000 server, the kernel extensions must be loaded. You must be root user to load the kernel extension:

```
/usr/lpp/adsmserv/bin/loadpkx -f /usr/lpp/adsmserv/bin/pkmonx
```

The kernel extensions need to be loaded only once after an initial boot of the system (IPL), and ADSM/6000 provides a script file *dsm_update_itab* in /usr/lpp/adsmserv/bin that puts the loading of the kernel into /etc/initab so that it starts automatically at boot time.

7.1.2 Estimating Database and Log Space

The administrator has to estimate the amount of database space and allocate this space for the database.

Estimation of the database space

The capacity required for the database is largely affected by the number of files that are backed up or archived.

To estimate the amount of space required for the database, first consider how many files you are going to back up or archive.

Our experience is the following:

1. Evaluate the number of files in each file system that you want to back up. For example, in a pre-existing system, you may issue the following command for each of the determined file systems, which counts the number of files it contains.

```
find /file_system_name  -print | wc -l
```

2. Multiply the number of files by 800 bytes. The space needed for the database is largely dependent on the size taken by each file in the database, and our experience is that each file needs about 800 bytes in the database.

3. Take the result of the preview operation and multiply it by the number of different backup versions you want the server to retain. You will specify later in the management class definition the number of versions.

4. Take a 20% to 25% margin.

5. Add 1MB for the activity log information.

   Because the activity log is stored in the database, the size of the activity log should be factored into the amount of space allocated for the database. According to the *ADSM/6000 Administrator's Guide* recommendations, you should determine the space for the activity log in respect to the number of files and the average number of active users. For large organizations, such as scientific and technical centers, we suggest a activity log space around 80MB.

Example

```
                                Small Center      Large Center
Number of defined users         =   50             2,000
Average number of files per user =  100               250
Number of files     (nfiles)     = 5000           500,000
Number of versions  (nvers)      =    2                 2

800 bytes * nfiles * nvers       =    8MB             800MB
8 MB * 0.25(margin)              =    2MB             200MB
Activity Log Space               =    1MB              80MB
=========================================================
Sum                                  11MB            1080MB
```

Estimation of the Log Space

The ADSM server maintains a recovery log, which keeps track of all changes made to the database if a system failure occurs. The recovery log is used to maintain the integrity of ADSM database.

If a system failure occurs, the server uses the information in the recovery log to roll back uncommitted transactions from the databases, so that the database remains consistent.

Based on our experience at the ITSO Open Systems Laboratory, we recommend that *Log space* should be 20MB for small and average centers (between 8% and 10% of the data base space). It is a starting point. The size of the recovery log is dependent on the number of concurrent client sessions communicating with the server, rather than the number of files in data storage. The number of concurrent sessions and the number of background processes executing on the server determine the number of transactions.

Initially, begin with 20MB for the recovery log. Then monitor the utilization of the recovery log to determine whether you should increase or decrease the size of the recovery log.

7.1.3 Defining AIX Logical Volumes

To allocate space on disk storage for the database and log, first set up logical volumes for the server. To do this, you must be logged on with root privileges. Create two logical volumes, using SMIT.

1. On the first panel, select **Physical & logical storage**.
2. On the next panel, select **Logical volume manager**.
3. On the next panel, select **Logical volumes**.
4. On the next panel, select **Add a logical volume**.
5. On the next panel, fill in the **Volume group name** field.

   You can select the volume group name using F4.
6. Provide a name in the logical volume name field.

   For example:

   ```
   For the ADSM database : “rdbvol1”
   For the ADSM Log      : “rlogvol1”
   ```
7. Fill in number of partitions.

   Give the number of partitions to reach the amount of space you have previously estimated.

7.1.4 Allocation of Space for Database and Log

To allocate space on disk storage for the database and the recovery log, submit the following command:

```
dsmserv install 1 /dev/rlogvol1 1 /dev/rdbvol1
```

Where *rlogvol1* and *rdbvol1* are the AIX volumes previously defined.

Now start the server to verify the database and the recovery log.

```
/usr/lpp/adsmserv/bin/dsmserv
```

To start it in background, add the -quiet option, such as:

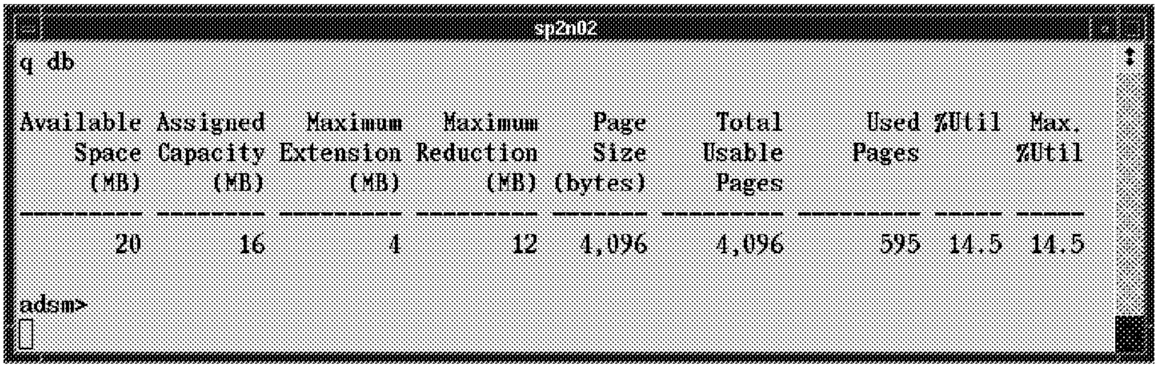
```
/usr/lpp/adsmserv/bin/dsmserv -quiet &
```

7.1.4.1 ADSM Database Verification

You can verify the amount of space available for the database use by entering:

```
adsm> query db
```

Enter this command on the server console interface. The server displays a report similar to Figure 10, which shows the amount of space available for database use.



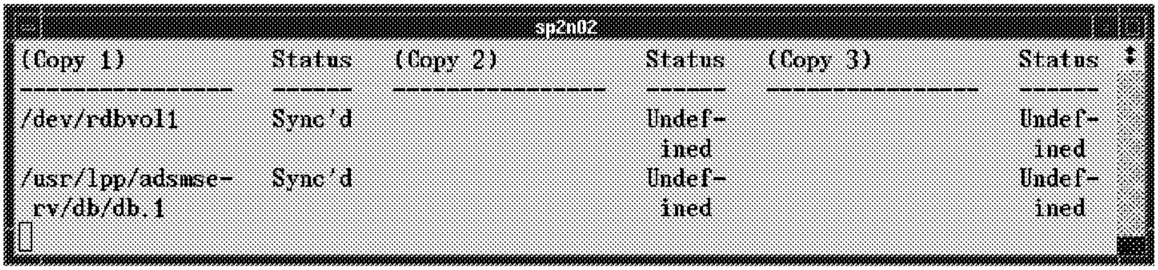
```
q db

Available Assigned   Maximum   Maximum    Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction    Size    Usable     Pages        %Util
     (MB)     (MB)      (MB)      (MB) (bytes)     Pages
--------- -------- --------- --------- ------- --------- --------- ----- -----
       20       16         4        12   4,096     4,096       595  14.5  14.5

adsm>
```

Figure 10. Database Information

You can also request information about the volumes assigned to the database by entering:

```
adsm> query dbvol
```

The server displays a report, similar to Figure 11. It shows all the volumes defined as database volumes.



```
(Copy 1)           Status   (Copy 2)           Status   (Copy 3)           Status
----------------   ------   ----------------   ------   ----------------   ------
/dev/rdbvol1       Sync'd                      Undef-                      Undef-
                                                ined                        ined
/usr/lpp/adsmse-   Sync'd                      Undef-                      Undef-
 rv/db/db.1                                     ined                        ined
```

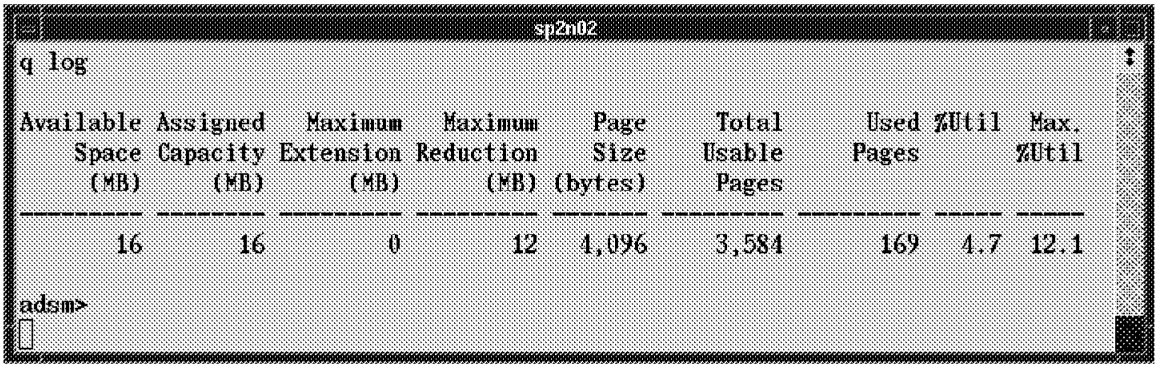
Figure 11. Database Volumes

7.1.4.2 ADSM Log Verification

You can verify the amount of space available for recovery log use by entering:

```
adsm> query log
```

The server displays a report similar to Figure 12, which displays the amount of space available for use for in the recovery log.



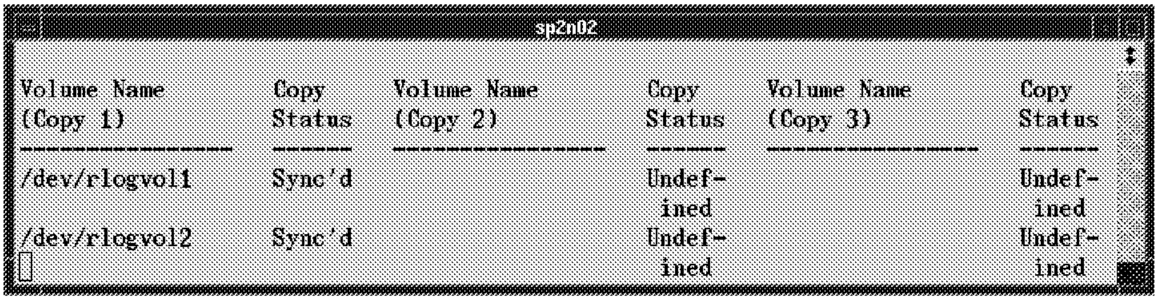
```
q log

Available Assigned   Maximum   Maximum     Page     Total      Used %Util  Max.
    Space Capacity Extension Reduction     Size    Usable     Pages       %Util
     (MB)     (MB)      (MB)      (MB)  (bytes)     Pages
--------- -------- --------- --------- -------- --------- --------- ----- -----

       16       16         0        12    4,096     3,584       169   4.7  12.1

adsm>
```

Figure 12. Log Information

You can also request information about the volumes assigned to the recovery log by entering:

```
adsm> query logvol
```

You can verify with this command, as shown in Figure 13, the AIX logical volumes defined as the recovery log.



```
Volume Name         Copy    Volume Name         Copy    Volume Name         Copy
(Copy 1)            Status  (Copy 2)            Status  (Copy 3)            Status
----------------    ------  ----------------    ------  ----------------    ------
/dev/rlogvol1       Sync'd                      Undef-                      Undef-
                                                 ined                        ined
/dev/rlogvol2       Sync'd                      Undef-                      Undef-
                                                 ined                        ined
```

Figure 13. AIX Volumes Defined to the Recovery Log

7.2 Storage Pool Environment

There are two storage pools, called BACKUPPOOL and ARCHIVPOOL, with a device class of DISK predefined with ADSM/6000. If you want to define additional storage pools, follow these instructions.

The procedure is not the same for disk and for tape. We will take the ITSO Open Systems Laboratory configuration for reference, and split this section into two topics:

- Defining storage pools on disk (IBM 9333 DASD)
- Defining storage pools on tape (IBM 3490E Model E11)

If you need more information about defining storage pools, refer to *ADSM/6000 Administrator's Reference*.

7.2.1 Storage Pool on Disk

The set up of a disk storage pool will be explained using the IBM 9333 DASD as reference. Verify that the IBM 9333 DASD is defined correctly and available in the AIX configuration. You can check it through SMIT by selecting:

```
Devices
Fixed disk
List all defined disks
```

7.2.1.1 Allocating Space

Create AIX logical volumes and define them as ADSM volumes for use in the disk storage pools. You must be logged on as root to create AIX logical volumes on the IBM 9333 DASD. Use the SMIT command and select:

1. **Physical & logical storage** on the first panel
2. **Logical volume manager** on the next panel
3. **Logical volumes** on the next panel
4. **Add a logical volume** on the next panel
   a. Fill the **volume group name** field.

      You can select a Volume Group Name for the IBM 9333 DASD using F4.

   b. Specify a name in the *logical volume name* field.

      For example:

      ```
      For the ADSM backup volume   : ″backol1″
      For the ADSM archive volume  : ″arcvol1″
      ```

   c. Provide the number of partitions.

      Specify the number of partitions to reach the amount of space you need for backup and archive.

7.2.1.2 Creating a Storage Pool on Disk

There are two ways to define the volumes in ADSM/6000:

- Using the command line interface, you must define the storage pool such as:

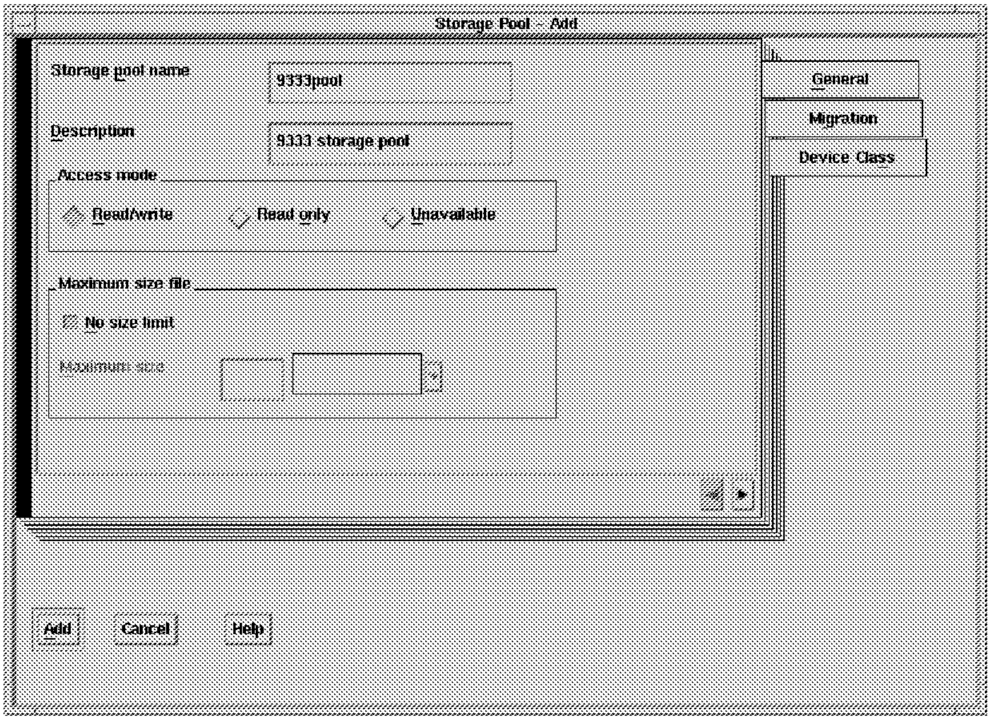
  ```
  adsm> define stgpool 9333pool disk
  ```

- Using the graphical interface you have to do the following steps:
  1. Start the graphical interface using the following command:

     ```
     dsmadm &
     ```

  2. Give your administrator user ID and password.
  3. Click on the **Data storage** icon.
  4. Select **Edit** option from the action bar.
  5. Select **Add** from the pull-down menu.

6. Give a name to your storage pool, as shown in Figure 14 on page 47.



Storage pool name: 9333pool

Description: 9333 storage pool

Access mode: ◆ Read/write ◇ Read only ◇ Unavailable

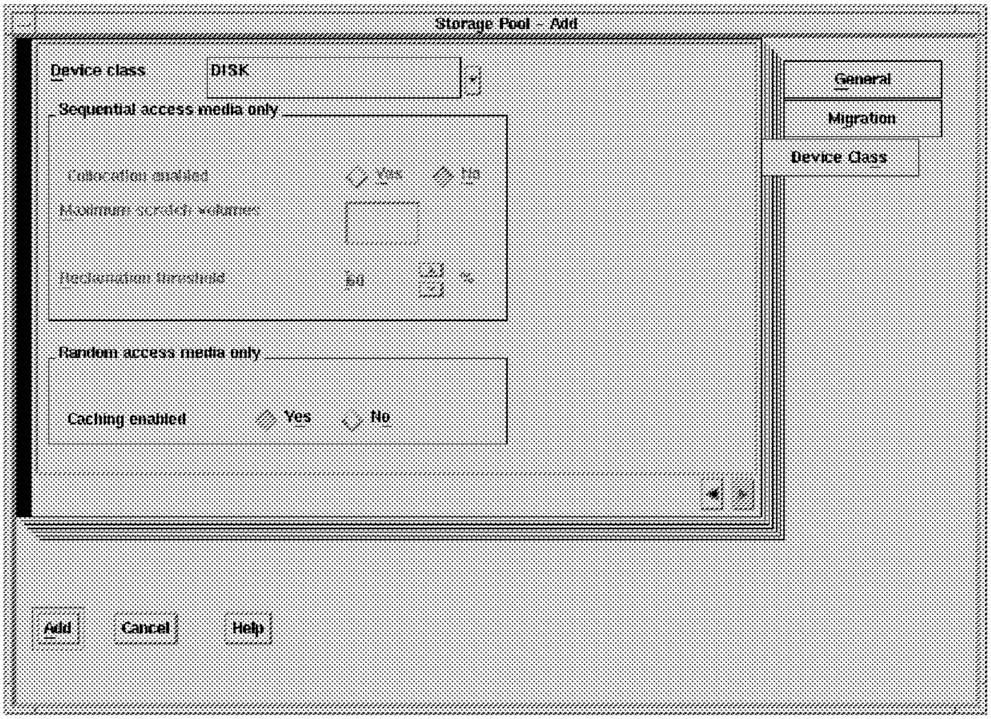
Maximum size file: ☒ No size limit

Maximum size

Figure 14. Storage Pool Creation with GUI, Part 1 of 2

7. On this panel, select the tab labeled **Device Class**

   On this panel, you must select the device class corresponding to the storage pool type you are creating. Here, you have to select **DISK**, as shown in Figure 15.



Device class: DISK

Sequential access media only

Collocation enabled ◇ Yes ◆ No

Maximum scratch volumes

Reclamation threshold 60 %

Random access media only

Caching enabled ◆ Yes ◇ No

Figure 15. Storage Pool Creation with GUI, Part 2 of 2

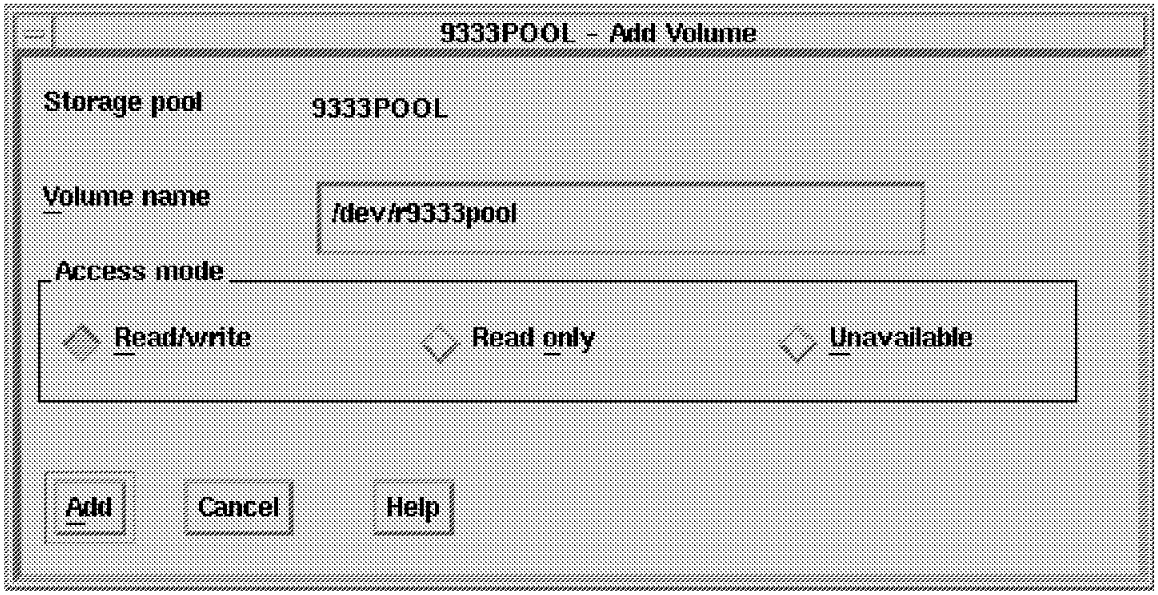
7.2.1.3 Assigning AIX Logical Volume to a Storage Pool

You must allocate to the storage pool the space (or logical volumes) you have created. Logical volumes can be assigned to ADSM storage pools with the command line interface or with the graphical interface (GUI).

- When using the command line interface, enter the following command, such as:

```
adsm> define volume 9333pool /dev/rstgvol1
```

- When using the graphical administrator interface, follow this procedure:
  1. Double click on the **data storage** icon.
  2. Double click on the **9333POOL** icon.
  3. Select **edit** on the option bar.
  4. Select **add** in the pull-down menu.
  5. Specify the name of the previously defined AIX logical volume, as shown in Figure 16.



9333POOL - Add Volume

Storage pool 9333POOL

Volume name /dev/r9333pool

Access mode

◆ Read/write ◇ Read only ◇ Unavailable

Figure 16. Adding a Volume to the Disk Storage Pool

7.2.2 Storage Pool on Tape

The set up of a tape storage pool will be explained using the IBM 3490E Model E11 as reference. Verify that the IBM 3490E Model E11 is defined correctly and available in your AIX configuration. You can check it through SMIT by selecting:

```
Devices
Tape drive
List all defined tape drive
```

If the tape is not defined, refer to *AIX Installation Instructions* to configure it. For the IBM 3490E Model E11, two device drivers are available:

- 3490E

- A3490E

Do not use the 3490E driver; ADSM will not work with this base device driver. Use the A3490E device driver.

7.2.2.1 Defining a Tape Library to ADSM/6000

When you define a storage pool, you specify the name of a device class associated with this storage pool. Each device class is associated with exactly one library through the LIBRARY parameter.

Since the IBM 3490E Model E11 is not a robot but only an autoloader, the library has to be defined with a type of manual.

```
adsm> define library 3490lib libtype=manual
```

7.2.2.2 Defining a Drive to ADSM/6000

To inform the server about a drive that can be used to access storage volumes, the administrator must issue the `define drive` command. Each drive is assigned to one library. In this command, you must specify the device name given by AIX when the tape has been configured:

```
adsm> define drive 3490lib 3490drv device=/dev/rmt0
```

7.2.2.3 Defining a Device Class to ADSM/6000

A device class represents a type of device that can be used to store backup and archive data. Each storage pool is associated with exactly one device class.

When a new device class is defined, its device type (devtype option) must be specified to indicate the type of device, and thus the type of storage volumes, that are used to store the files. For the IBM 3490E Model E11, you must specify *cartridge* for the device type. You can also specify the recording format used by ADSM when writing data to a tape (FORMAT parameter). For the IBM 3490E Model E11, we used the 3490C format options to specify a 36-track compacted recording format.

```
adsm> define devclass 3490dev devtype=cartridge format=3490C library=3490lib
```

7.2.2.4 Creating Storage Pools for the Tape

To define your IBM 3490E Model E11 as a storage pool, you can use either the command line interface or the graphical interface.

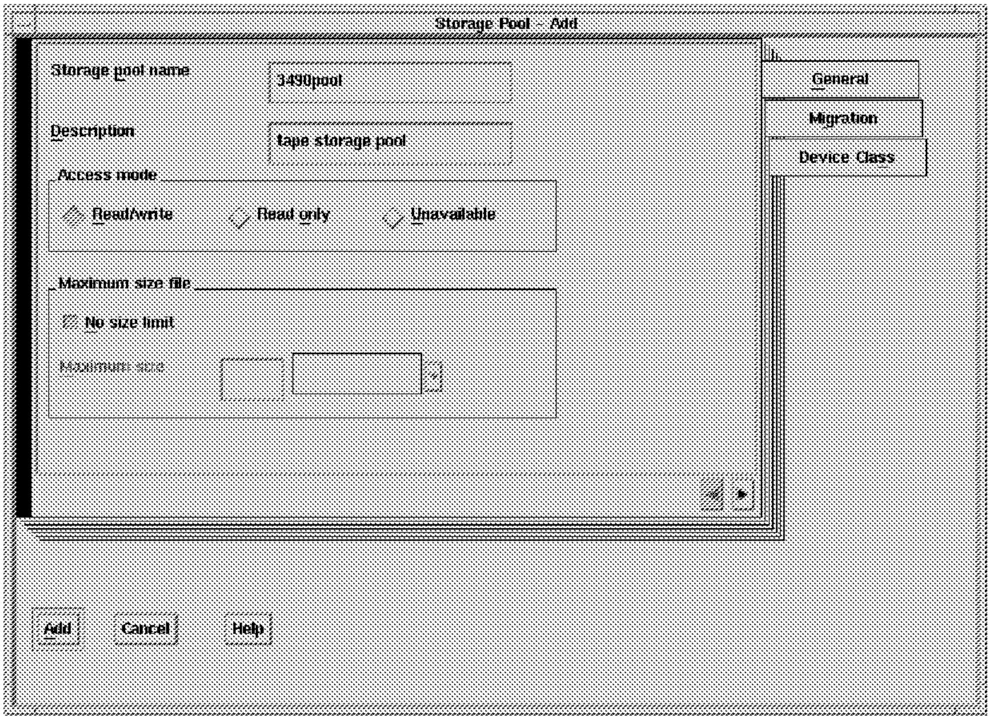
- If you use the command line interface of the server, you must enter a command such as:

  ```
  adsm> define stgpool 3490pool 3490class maxscratch=50
  ```

  Specify a nonzero value for the MAXSCRATCH option if you want to use scratch volumes. The use of scratch volumes is recommended with tape storage pools. When tapes are explicitly defined as volumes in ADSM/6000, ADSM will ask you to mount specific volumes, which is undesirable if you use the Cartridge Stack Loader of the IBM 3490E Model E11. Scratch volumes are not predefined as volumes in ADSM. Whenever ADSM requires space in a storage pool and the MAXSCRATCH number of tapes has not been exceeded, ADSM accepts any scratch tape, and ADSM defines it as a volume automatically when it starts to use it.

- If you use the graphical interface, do the following steps:

  1. Start the graphical interface (dsmadm program).
  2. Click on the **data storage** icon.
  3. Select **edit** option from the action bar.
  4. select **add** from the pull-down menu.
  5. Give a name to your storage pool, as shown in Figure 17.



Storage Pool - Add

Storage pool name: 3490pool

Description: tape storage pool

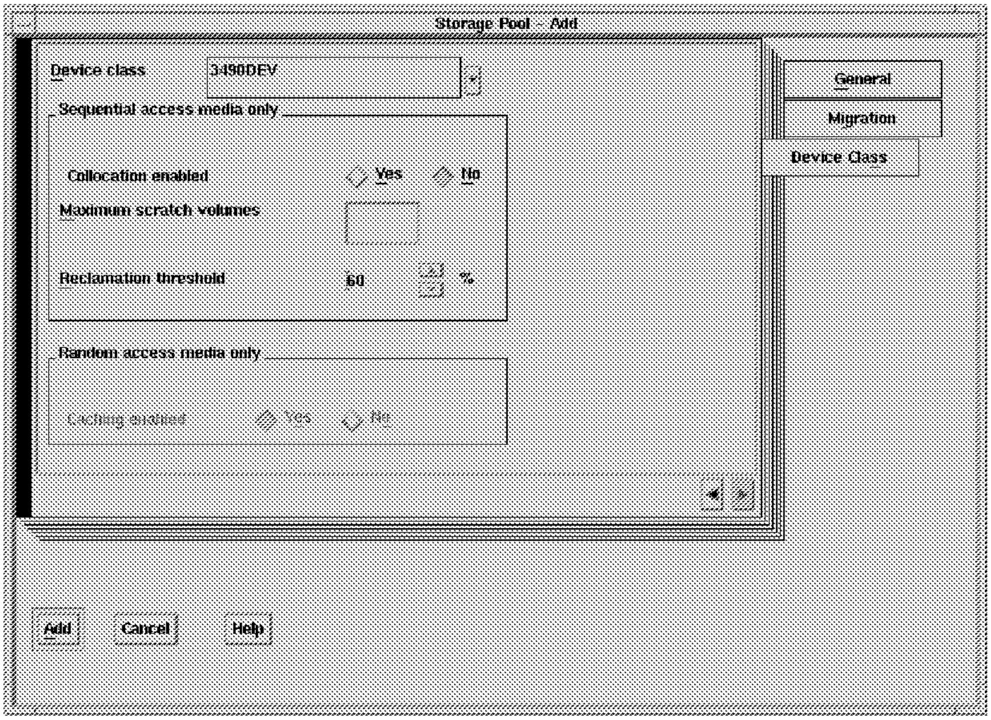
Access mode: ◆ Read/write ◇ Read only ◇ Unavailable

Maximum size file: ☒ No size limit

Maximum size

Figure 17. Tape Storage Pool Creation, Part 1 of 2

  6. On this panel, select the tab labeled **Device Class**. On the next panel you must select the device class corresponding to the storage pool type you are creating. Here, you have to select **3490DEV**, as shown in Figure 18 on page 51.



Device class 3490DEV

Reclamation threshold 60 %

Figure 18. Tape Storage Pool Creation, Part 2 of 2

7.2.2.5 Labelling Tapes

Tapes are considered volumes in ADSM. You need to label each tape or cartridge before use with ADSM.

1. Load six new cartridges in the IBM 3490E Model E11.
2. Use the dsmlabel command. To label the cartridges that you have just loaded, use the dsmlabel command. This command is located in /usr/lpp/adsmserv/bin. The following command has to be executed from the AIX prompt:

   ```
   dsmlabel -drive=/dev/rmt0
   ```

   You may want to look at 11.3, “Using the IBM 3490E Model E11” on page 94 for additional information on how to operate the IBM 3490E Model E11.

7.3 Basic Administration Tasks

The last steps before executing the first backup are:

1. Define an ADSM/6000 administrator.
2. Specify a name for the ADSM/6000 server.
3. Register the license.
4. Register all ADSM/6000 clients.

Each ADSM/6000 client must be registered as a node in the ADSM/6000 server database before the client can perform any ADSM operation.

7.3.1 Register a New Administrator

After the installation process, register at least one administrator and grant the administrator system authority:

1. Register the new administrator entering the command below on the ADSM server console:

```
adsm> register admin adminame password
```

2. Use the GRANT command to grant this administrator the system privilege class:

```
adsm> grant authority adminame classes=system
```

7.3.2 Naming the Server

To identify the server, name the server. You should specify the name by which the server is to be identified by the client. This name is displayed to any client that accesses the server.

We suggest using the name you have defined in your client options files. Do this by entering the following command on the server interface:

```
set servername sp2sw02
```

If you don′t change the name, the default name is ADSM.

7.3.3 License Registration

A license must be registered with ADSM/6000 before nodes can be registered. To be precise, the default license will allow you to register 1 node and use 25GB, but additional node registrations are denied by the license manager.

The license for ADSM/6000 has to be requested from IBM, who returns the license number by fax or mail to the customer. There are two ways to establish the license: one is to edit the file /usr/lib/netls/conf/nodelock, the other is to use the ADSM/6000 command *register license*. If you edit the file /usr/lib/netls/conf/nodelock, you have to enter a line that looks like the following:

```
 63d9e18fb3cb.02.09.73.01.02.00.00.00 g4vjjf5839d2hwbpvtje3cvphwja ″25GB,25CL″ ″1.2″
```

If you use the *register license* command in ADSM/6000, you need to enter a command, such as:

```
adsm> register license g4vjjf5839d2hwbpvtje3cvphwja 25GB,25CL
```

Note: The notation is case sensitive. Use exactly the case notation that has been given to you from IBM.

7.3.4 Nodes Registration

Before an SP2 user can begin requesting backup or archive services from an ADSM server, the SP2 nodes have to be registered with a server.

ADSM provides two methods for registering nodes:

- Open registration

  With open registration, root users can register their workstations with the ADSM server. Using this method, the user will receive a prompt at his first connection to the server asking him to declare his node name and password. Open registration has the disadvantage that only a root user can register a workstation as a client with the ADSM/6000 server. If the ADSM/6000 administrator has to login to each SP2 node to register the node as an ADSM/6000 client, the closed registration is a better choice.

- Closed registration

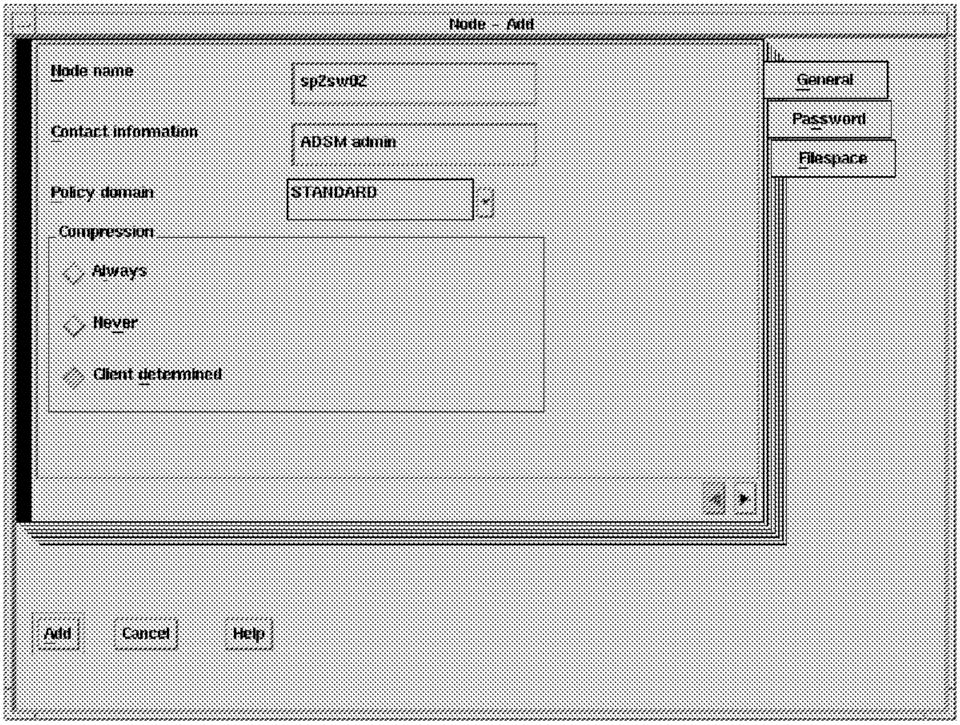
  The administrator must register each SP2 node as an ADSM/6000 client. Closed registration requires you to assign the node name, the node password, and the policy domain to which the ADSM/6000 client is assigned. If there are many SP2 nodes that need to be registered, think of using an ADSM macro to perform this task. The use of ADSM macros is explained in *ADSM/6000 Administrator Reference*. There are also some examples in 11.5, “Using ADSM Macros” on page 98.

Registration can be done at the command line interface of the server, such as:

```
register node sp2sw14 pswd14
```

To register nodes using the graphical interface, follow these steps:

1. Double click on the **node** icon.
2. On the next panel, select **edit** on the action bar.
3. Choose the **add** option in the pull-down.



Node name: sp2sw02

Contact information: ADSM admin

Policy domain: STANDARD

Compression: ◆ Client determined

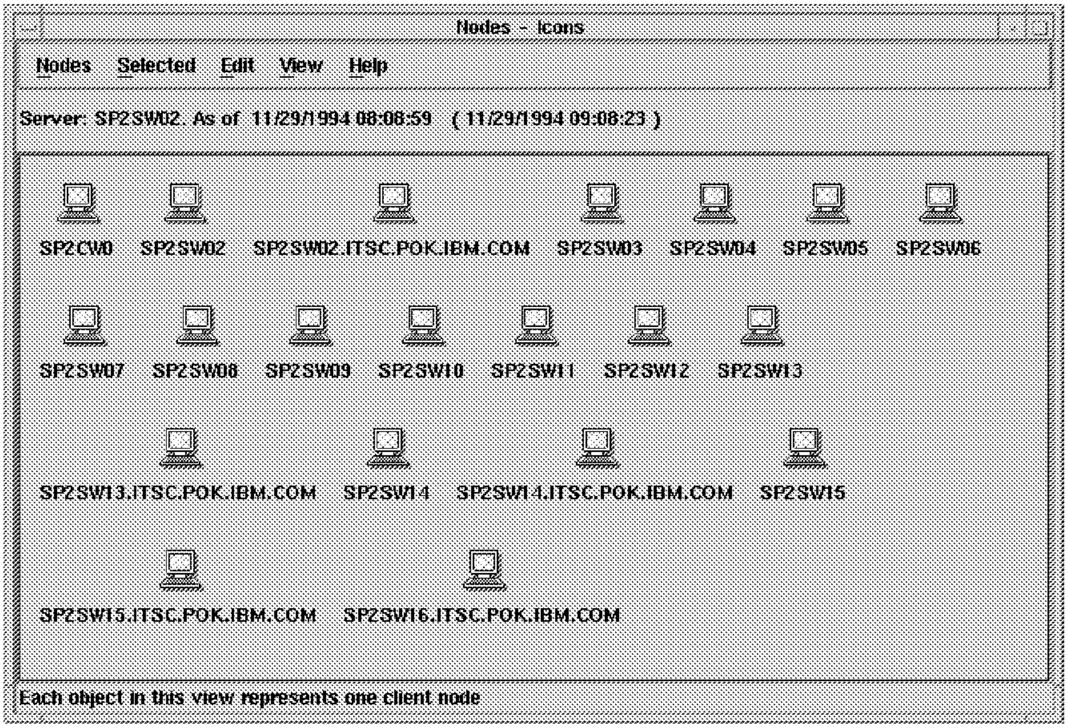
Figure 19. Nodes Registration

On this panel, you must:

- Name the node
- Select a policy domain
- Give a password to the node

4. To initialize the registration, click on the **Add** push button.

When your nodes are defined using either the closed registration or the open registration, you can visualize them as shown in Figure 20 on page 55, selecting the *node* icon on your administrative interface.



Server: SP2SW02. As of 11/29/1994 08:08:59 (11/29/1994 09:08:23)

Each object in this view represents one client node

Figure 20. Nodes Defined

Section 8. ADSM Verifications

The ADSM/6000 server and client are now installed and configured. Before going further, you can do some verifications to be sure that this first part is well customized:

- Verify the communication between your client and the server by starting the user interface.
- Verify the storage pool definition by backing up a small file on each storage pool.

8.1.1 Communication Verification

We suggest first to verify the communication between the server and the client. An easy way to verify it is to run the user graphical interface (DSM program) from the client. When starting, this program will try to establish the communication between the client and the server.

To start it, enter the following command:

```
dsm &
```

If it doesn′t work, we have noticed three possible reasons:

Communication parameters error

Your client cannot establish the communication with the server. You can have this problem if:

- There is a wrong address in the client option file.

  Verify the address in the dsm.sys option file.

- You changed the default communication port.

  Verify in the client dsm.sys file and in the server dsm.opt file that the communication port is the same.

No license available for client access

The pricing model being established for ADSM is based on total server storage usage and the number of registered client nodes that can connect to the server. License terms are specified on the server with iFOR/LS (formerly NetLS) licenses obtained from IBM Customer Support. If you didn′t ask for a iFOR/LS (formerly NetLS) key or if you made a mistake creating the *nodelock* netls file, you get a message saying that the number of licensed users is exceeded.

To verify how many licenses are available, use the QUERY LICENCE command.

Your client is unknown

- If you have used closed registration:

  You named the node using the register command or the administrator GUI. By default, the node name must be the node′s hostname. You can register another name, provided that the client is initialized with a customized *dsm.opt* file with the *nodename* parameter set up to the chosen client name.

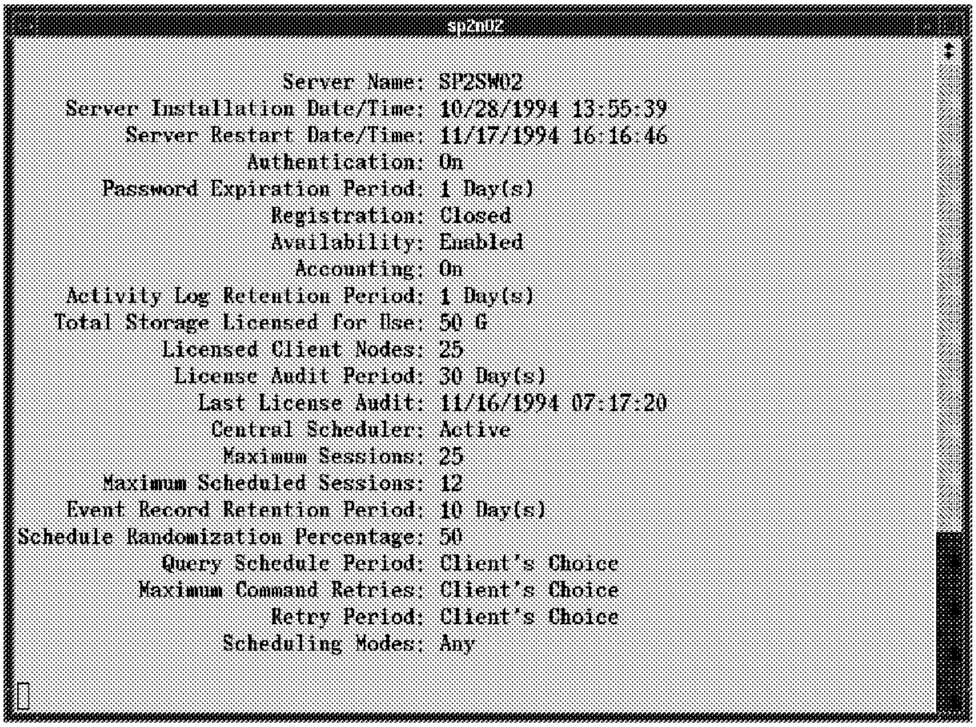
- If you have used open registration:

  At the very first connection to the server, ADSM will ask you to register your node giving its name and password. As described in the closed registration paragraph, either you use the nodename option or you give the node′s hostname.

To verify the registration mode, use the following command on the server console:

```
query status
```

You will get the information shown in Figure 21.



```
                      Server Name: SP2SW02
   Server Installation Date/Time: 10/28/1994 13:55:39
        Server Restart Date/Time: 11/17/1994 16:16:46
                  Authentication: On
       Password Expiration Period: 1 Day(s)
                    Registration: Closed
                    Availability: Enabled
                      Accounting: On
    Activity Log Retention Period: 1 Day(s)
   Total Storage Licensed for Use: 50 G
           Licensed Client Nodes: 25
            License Audit Period: 30 Day(s)
              Last License Audit: 11/16/1994 07:17:20
               Central Scheduler: Active
                Maximum Sessions: 25
       Maximum Scheduled Sessions: 12
    Event Record Retention Period: 10 Day(s)
Schedule Randomization Percentage: 50
           Query Schedule Period: Client's Choice
         Maximum Command Retries: Client's Choice
                    Retry Period: Client's Choice
                Scheduling Modes: Any
```

Figure 21. Query Status Command

Warning: Be careful that the *nodename* and *password generate* options do not work together. If you use the *nodename* option, ADSM prompts you for the password whether the *password generate* option is specified or not.

8.1.2 Storage Pool Verification

One way to verify that your storage pools are well defined is to execute a simple backup on each storage pool.

Before executing the backup, you can customize your default management class. After ADSM installation, a STANDARD policy domain is created. This STANDARD policy domain has a STANDARD management class with a backup copy group, and an archive copy group. These copy groups indicate on which storage pool files will be stored.

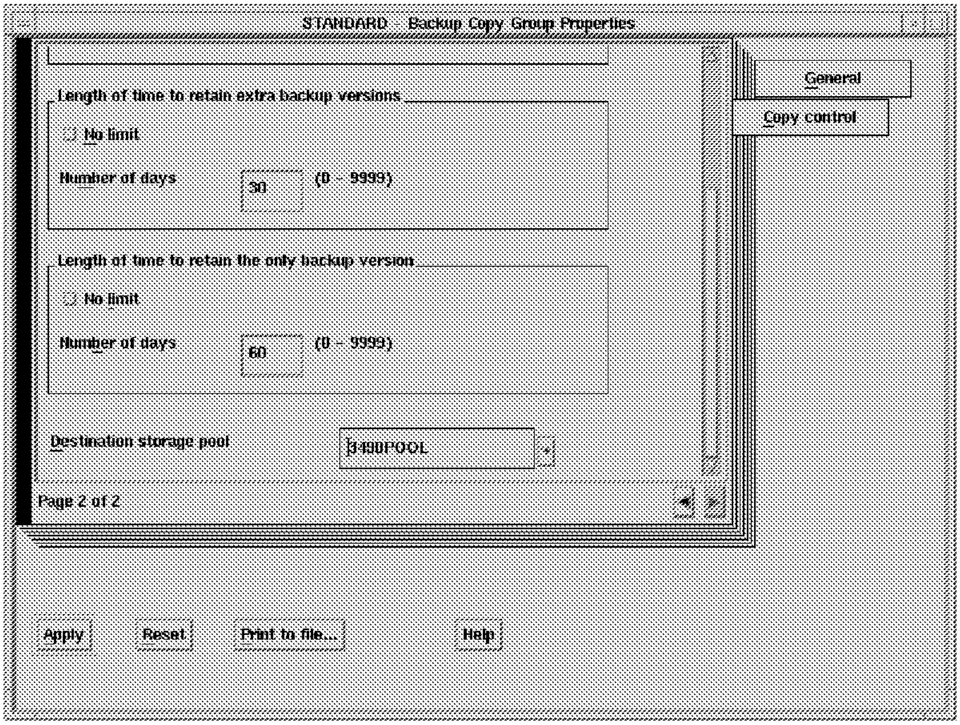
Defining the storage pool destination

If you want to test your IBM 3490E Model E11 storage pool, change the destination of your backup in the backup copy group. To do this:

- Using the graphical interface:

  1. Double click on the **Policy** icon.
  2. On the policy domain panel, double click on the **Standard** icon.
  3. On the policy set panel, double click on the **Standard** icon.
  4. On the management class panel, double click on **Standard** icon.
  5. On the copy group panel, click on the **Backup** copy group and chose the **Selected** option.
  6. In the pull-down menu, select **Open as properties**.
  7. Select **Copy control** option shown on Figure 22. The next window is displayed, which allows you to change the destination storage pool by clicking on the combo box.
  8. You must come back to the management class panel, click on the **Standard** policy set, and use the policy set menu to activate it.

     This operation is necessary to update the active policy set with your last modification.



Length of time to retain extra backup versions

□ No limit

Number of days 30 (0 - 9999)

Length of time to retain the only backup version

□ No limit

Number of days 60 (0 - 9999)

Destination storage pool 3490POOL

Figure 22. Modifying the Storage Pool Destination

- Using the server command line you can update the storage pool destination parameter with the following command:

```
adsm> update copy group STANDARD STANDARD STANDARD type=backup destination=3490pool
```

  Then you must activate the STANDARD policy set using the following command:

```
adsm> activate policyset standard standard
```

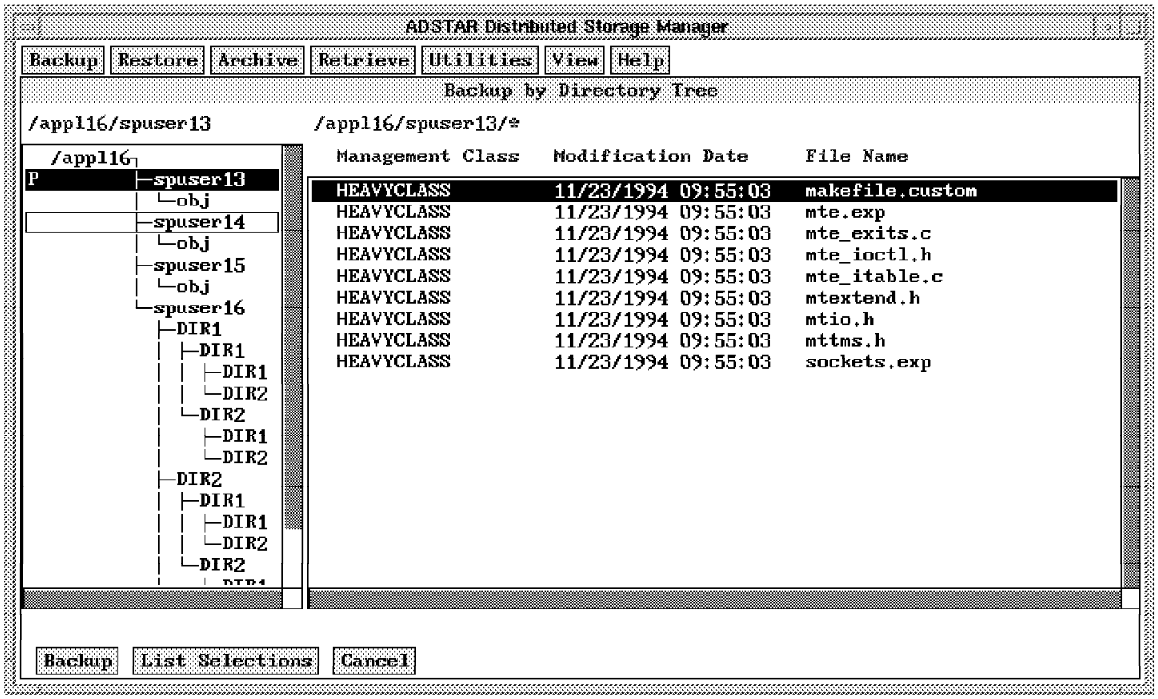
Executing a simple backup

You can execute a backup to the new destination using the graphical interface:

1. Start the user interface using the following command:

```
dsm &
```

2. Select the file system you want to backup.
3. Select **Backup** in the action bar.
4. Select **by Directory Tree** in the pulldown menu.
5. Click on the directory or the file you want to backup.
6. Execute the backup by clicking on the **Backup** pushbutton.



/appl16/spuser13

```
 /appl16┐
P       ├─spuser13
        │ └─obj
        ├─spuser14
        │ └─obj
        ├─spuser15
        │ └─obj
        └─spuser16
          ├─DIR1
          │ ├─DIR1
          │ │ ├─DIR1
          │ │ └─DIR2
          │ └─DIR2
          │   ├─DIR1
          │   └─DIR2
          ├─DIR2
          │ ├─DIR1
          │ │ ├─DIR1
          │ │ └─DIR2
          │ └─DIR2
```

/appl16/spuser13/*

```
Management Class   Modification Date     File Name
HEAVYCLASS         11/23/1994 09:55:03   makefile.custom
HEAVYCLASS         11/23/1994 09:55:03   mte.exp
HEAVYCLASS         11/23/1994 09:55:03   mte_exits.c
HEAVYCLASS         11/23/1994 09:55:03   mte_ioctl.h
HEAVYCLASS         11/23/1994 09:55:03   mte_itable.c
HEAVYCLASS         11/23/1994 09:55:03   mtextend.h
HEAVYCLASS         11/23/1994 09:55:03   mtio.h
HEAVYCLASS         11/23/1994 09:55:03   mttms.h
HEAVYCLASS         11/23/1994 09:55:03   sockets.exp
```

Figure 23. Executing a Backup

Using the command line, the backup would look like:

```
dsmc query backup filename
```

Verifying the content of the storage pool

You can verify that your file or directory has been well backed up by printing the contents of the storage pool volumes to a file. For example, if you have backed up files on TAPE02 belonging to the storage pool 3490POOL, follow these steps:

1. On the administration interface, double click on **Data Storage**.
2. On the next panel, double click on the storage pool **3490POOL**.
3. On the next panel, select **TAPE02**.
4. Choose the **Selected** option in the action bar.

5. Select **List Volume Content to File**.

You will obtain a file similar to the following.

```
TAPE09 - List Volume Content to File
Node Name Type File Space  Client File name     Size of   Part of File
                                              stored file stored on this
                                                          vol/total part

SP2SW15  Bkup /appl15  /spuser15/bank/ README       2690      1/1
SP2SW15  Bkup /appl15  /spuser15/bank/ bank.idl     1942      1/1
SP2SW15  Bkup /appl15  /spuser15/bank/ bankd.c     10146      1/1
SP2SW15  Bkup /appl15  /spuser15/bank/ baybanks      107      1/1
SP2SW15  Bkup /appl15  /spuser15/bank/ shawmut       201      1/1
SP2SW15  Bkup /appl15  /spuser15/bank/ util.c       1792      1/1
SP2SW15  Bkup /appl15  /spuser15/bank/ util.h        622      1/1
SP2SW15  Bkup /appl15  /spuser15/bank/ uuidbind.c   3644      1/1
SP2SW15  Bkup /appl15  /spuser15/binop_c/ binop.idl  791      1/1
SP2SW15  Bkup /appl15  /spuser15/binop_c/ client.c  2172      1/1
SP2SW15  Bkup /appl15  /spuser15/binop_c/ server.c  1695      1/1
SP2SW15  Bkup /appl15  /spuser15/perf/ client      46000      1/1
SP2SW15  Bkup /appl15  /spuser15/perf/ run_client   2397      1/1
SP2SW15  Bkup /appl15  /spuser15/perf/ server      25607      1/1
SP2SW15  Bkup /appl15  /spuser16/bank/ Makefile      990      1/1
SP2SW15  Bkup /appl15  /spuser16/bank/ bank.c       4833      1/1
SP2SW15  Bkup /appl15  /spuser16/bank/ bank.idl     1942      1/1
SP2SW14  Bkup /appl14  /spuser13/ 3270keys.hft      4921      1/1
SP2SW14  Bkup /appl14  /spuser13/ addrs.awk         2860      1/1
SP2SW14  Bkup /appl14  /spuser13/ anon.ftp          4614      1/1
SP2SW14  Bkup /appl14  /spuser13/ gateways          2847      1/1
```

Chapter 3. Scenarios

This chapter provides a description of the scenarios we have defined. We describe two different scenarios aimed respectively at scientific/technical users (hereafter named S/T users) and commercial users. The scenario for S/T users has also been set up and tested on the ITSO Open Systems Lab. IBM 9076 SP2.

The user requirements from the two environments are very different as are the user expectations. Hence, the present chapter is split into two sections each dealing with one of the environments. In each section we first discuss the kind of users we can expect in that environment along with their needs of computer resources.

Then we define sets of data that emulate the typical data distribution for that environment. Last, we define one or more ADSM configurations that can be effectively used to back up and archive the user′s data.

Section 9. Scientific and Technical Scenario

The scientific and technical (S/T) users may have different storage requirements according to their main activity. In the next section we outline some of the most relevant scientific and technical activities and their requirements in terms of disk usage, access speed and system reliability. In the second section we sketch a set of scenarios to emulate different scientific environments (like universities and academic institutions) or technical environments (like industries). In the last sections, we suggest ADSM configurations that can be used to provide the backup/archive service to SP2 S/T users. We address both system-initiated operations and user-initiated operations.

In this scenario we assume the customer has a hardware configuration such as shown in Figure 24.

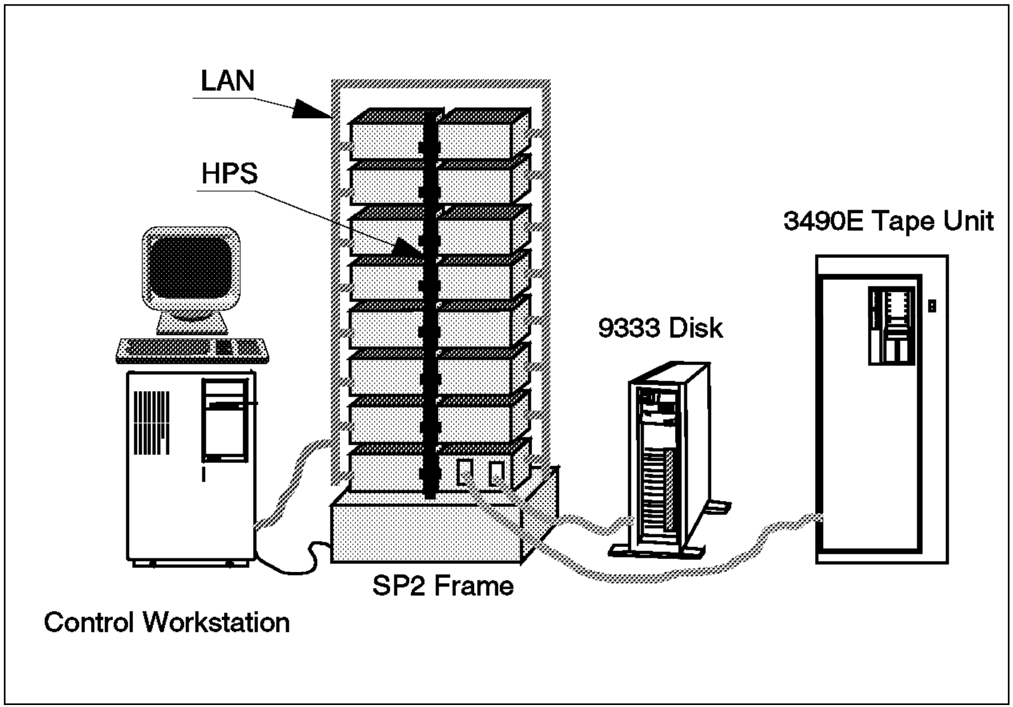


Figure 24. Hardware Configuration of S/T Environment

9.1 User Requirements

We sketch some categories of S/T users. We do not pretend to offer an exhaustive description of all users. We believe that most users will fit into one or more of those categories. We also believe that users who do not fit any category do not invalidate our scenarios as the scenarios do cover a wide range of file sizes, number of files, reliability requirements and speeds of access. The next several subtopics describe some classes of S/T users along with their requirements.

9.1.1 Programmers

The typical activity of a programmer is a cycle. The programmer starts writing some code and performing small tests to check the functionality of the code. In that phase, many new small files can be created (both source and object files) and they are frequently modified. When the code writing phase is close to completion, the programmer starts more extensive testing and debugging. He needs more disk space as executables and libraries get larger and larger. In that phase, files are still frequently created and updated. When the code development is finished, the last phase of the cycle starts. Files do not grow any more in number and size and are updated less frequently. The last phase can last longer than the other phases.

The source files are by far the most important files to a programmer. Typically, they are numerous and small, often spread over many directories. Their typical size is a few KBytes, and they can be many hundreds. The typical access to source files is for editing, so the speed of access is not really important. However, those files are extremely important in that essentially all of the programmer′s work is stored in them.

The object files are much less important, as they can be easily recreated by recompiling. However, the compile phase can take such a long time (tens of hours) that the programmer might also wish to save object files. The object files are typically numerous and small, often spread over many directories just like source files. Often, they are organized with a directory hierarchy that closely resembles the source files directory hierarchy. The compilation phase is usually CPU bound, so a relatively slow file access can be accepted, although it will for sure negatively affect the compilation time.

Executables and libraries can be significantly larger than object or source files (they can be up to tens of MBytes). However, provided the object files still exist, they are relatively fast to recreate. They are surely the less important files to programmers. The linking phase is usually I/O bound, so file access speed is important.

9.1.2 End Users

The end users of S/T systems typically require much CPU power. However, storage requirements can vary significantly and are often higher for technical (industrial) users than for scientific (academic) users. Many S/T applications are quite large and complex, especially industrial ones, and the file size for executables is often in the range of tens of MBytes. We can split the S/T end users into two subclasses, according to the average requirement for storage space:

- Users who run code with relatively small input and output files (typically less than 10MB). These users are very frequently in the academic environment as well in the industrial environment and often use home-written code. The input and output files usually reside in the user′s working directories, while the codes can be located in the user′s working directories as well as in some system directory (for example, /usr/lpp or /usr/local/bin). These users want to backup or archive only the input and output files but expect also to be able to quickly reinstall the application in case of system failure.
- Users who run code with large input and output files (typically hundreds of MBytes, sometimes up to 2GB). Those users are more frequently in the industrial environment. The input and output files usually reside in the

user′s working directories, while the code is usually installed in a system directory (typically /usr/lpp). Those users want to back up only the input and output files. They also expect the application to be readily available after a system failure.

9.2 NFS and JFS Data

From the previous discussion, we can classify data storage for scientific and technical users into two broad classes:

- Complex trees of directories including many small files
- Simple trees of directories including a few large files

In most cases we can expect the first class to access the data through NFS from one or more servers. The second class is likely to use the local disk for performance reasons. Of course, that introduces asymmetries in the SP2 configuration and makes system management more complex. Nevertheless, performance reasons can force that choice.

9.2.1 Programmers

To satisfy the user′s requirements and expectations stated in the previous section, the system manager will locate user files in the following fashion:

- Source files can always be accessed through NFS: they are usually edited (editing is typically so slow that NFS speed is largely sufficient) or read by the compiler (NFS performance in reading a small file is acceptable and compilation is usually CPU bound). That leads to a complex tree of small files accessed through NFS.
- Object files could be either local or NFS mounted depending on their size and number at each compilation step. That leads to a complex tree of small files, either accessed through NFS or local to the node. However, for sake of simplicity, they are often accessed through NFS from the same server where the source files reside.
- Executable files should be local, as the speed of the linking procedure depends essentially on the speed of file access, and NFS performance in writing large files is quite poor. However, again for the sake of simplicity and to make executable files accessible from any node, it is preferable to have the link procedure work on a local file. It can then be copied to the NFS server where the source and object files reside.

In short, usually all programmer′s files should be located on the NFS file server, and they usually are in a complex directory tree including many small files and a few large files.

9.2.2 End Users

As seen, we can split the S/T end users into two classes, according to the average requirement for storage space:

- Users who run code with small input and output files (typically less than 10MB)
- Users who run code with large input and output files (typically hundreds of MBytes, sometimes up to 2GB)

For performance reasons, users from the second class need to access their data from the local disk. Users from the first class can sometimes have their data accessed through NFS, depending on their needs in terms of performance.

9.2.3 Simulation

A good compromise to simulate the S/T environment is to have two different storage environments:

- An NFS mounted file system containing a complex tree of many small files. It simulates the sources and objects file tree as well as the file storage for users with limited space requirements.
- A local (JFS) file system containing a simple tree of a few large files. It simulates users with large space requirements.

That also allows you to interpolate somehow the configuration and performance data for users with intermediate needs in term of storage space, directory complexity and speed of access.

For performance reasons, as already discussed, both kinds of file systems should be saved by the ADSM client running on the same machine where the files physically reside. It can either be the node itself or an NFS server.

9.3 NFS Data Backup

In the ITSO Open Systems Lab.′s setup, users are allowed to run their applications on nodes 3 to 12. Data are accessed through NFS from an NFS file server. The server can either be another node or the SP2 control workstation. Each of these nodes will need to run the ADSM/6000 client. However, since they all share the /usr file system from the SP2 control workstation, the client code must be installed only on the SP2 control workstation. All nodes can execute the ADSM/6000 client code. Some minor local customization should be done through symbolic links from /usr/lpp/adsm/bin to /var/adsm, as described in Section 7, “ADSM Basic Customization” on page 41.

Node 3 to Node 12

```
local       /(root)
shared      /usr    from the SP2 control workstation
local       /tmp
local       /var
shared      /home   from the SP2 control workstation
```

The approach we follow is that the regular backup work is done from the NFS servers themselves. This has the advantage that the data does not have to travel twice across the network, once from the NFS server to the node and then from the node to the ADSM/6000 server. Instead, each NFS server will directly back up its local data to the ADSM/6000 server. This way the data can be handed to the ADSM/6000 server faster. We have chosen to store the files to disk instead of tape for the following reasons:

- The space occupied is relatively small.
- The disk is slower than the tape to read or write large amounts of data, so we should reserve the tape for larger files.
- The disk is faster than the tape to retrieve files.

9.4 JFS Data Backup

The most I/O intensive applications in our scenario run on nodes 13 to 16. Each node has one or more local file systems to keep the application data.

Node 13 to Node 16

```
local     /(root)
shared    /usr  from Control Workstation
local     /tmp
local     /var
local     /home
local     /appl13
```

All nodes access the ADSM/6000 client from the shared /usr file system on the control workstation. So there is no need to install the ADSM/6000 client on each of these nodes. This topic has already been discussed in the previous section. Nodes 13 to 16 have most of their data on local file systems. These files have to be backed up by the nodes and will go to tape immediately for the following reasons:

- The tape is faster than the disk to sequentially read or write long files.
- The tape is slower than the disk to access a file, but the access time is much less important for very large files.

9.5 ADSM Server Setup

The ADSM/6000 server runs on node02. This node has access to 8GB of external disk, and to the IBM 3490E Model E11.

The ADSM/6000 server has three storage pools:

- 9333POOL on the IBM 9333 DASD
- BACKUPPOOL on the internal DASD
- 3490POOL on the IBM 3490E Model E11

These criteria are shown in Figure 25.

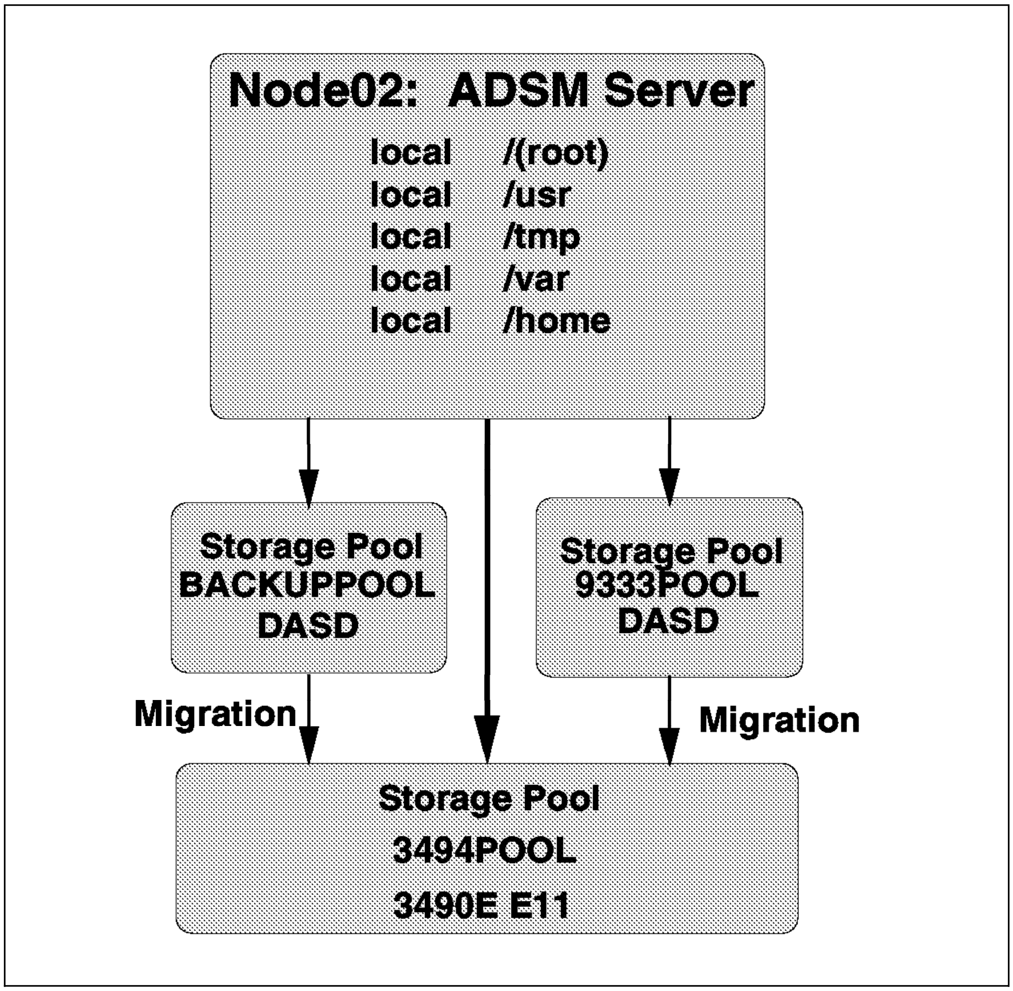


Figure 25. ADSM Server Setup for S/T Scenario

Table 1 on page 71 summarizes the classes of users we have defined along with the policies and schedules we have set up on the server.

]

Table 1. S/T Users Storage

User Type	Data Type	File Size	# of Files	File System	Directory	Management Class	Schedule	Client nodes served
Programmers	Sources	10KB	Around 100	NFS	/u/*user*	LightClass 1	Incremental every night	Control Workstation, node 2
	Objects	10KB	Around 100					
	Executables	100KB	< 10					
End Users	Small files	< 10MB	< 100			HeavyClass1 2		
	Large files	> 10MB	< 10	JFS	/appl<x> 4	HeavyClass2 3	Incremental every week	Node 13 to 16

Note:

1 LightClass: Backup/Archive to IBM 9333 DASD

2 HeavyClass1: Backup/Archive to IBM 9333 DASD and migrate to IBM 3490E Model E11

3 HeavyClass2: Backup/Archive to IBM 3490E Model E11

4 /appl13, /appl14, /appl15, and /appl16 are local file systems with application related data.

Section 10. Commercial Scenario

A commercial scenario on the IBM POWERparallel Systems SP2 is very likely to deal with transaction processing and database handling. It is obvious that there is a strong need to find solutions for backup and recovery of databases. The backup process for databases must work while the database is online and must finish in a reasonable amount of time, which is not easy considering that databases can easily be 50GB or more. Solutions for database backup with ADSM/6000 are outlined in *Using ADSM to Back Up Databases*. This section does not deal with database backup solutions.

However, there are other business solutions that run on the IBM POWERparallel Systems SP2. In fact, all business applications available for IBM AIX/6000 are in some way usable on the IBM POWERparallel Systems SP2 when installed on a single node.

The scenario in this section outlines a system with the following application software installed:

- Financial and business applications
- Publishing application
- Graphic design application
- Office application

We will show what backup/archive procedures could be implemented for such an environment, and provide information about the ADSM/6000 customization required for this operation.

Let us assume the customer uses hardware such as outlined in Figure 26 on page 74.

Notice, that the IBM 3494 Tape Library is connected to both frames, node00 and node16. This is possible because the IBM 3494 Tape Library has two attachments for each control unit. The robotic device to mount the tapes is shared between node00 and node16; however, the drives are not. One drive is assigned to node00, the other is assigned to node16. The tapes in the library are also not shared; they belong either to a category that is available on node00 or to another category that is available to node16.

The first frame has an IBM 7135 RAIDiant Array and an IBM 3995 Optical Library Dataserver storage device attached; the second frame has an IBM 9333 DASD storage device attached. These devices are all used for the backup/archive functions of this scenario. There may be more storage devices attached for application data, for example an external NFS file server.

ADSM Server 1

- ADSM Server 1 runs on node00 (wide node).
- ADSM Server 1 serves all applications running in frame 1.
- ADSM Server 1 has access to:
  - IBM 3995 Optical Library Dataserver
  - IBM 3494 Tape Library with 240 cartridges

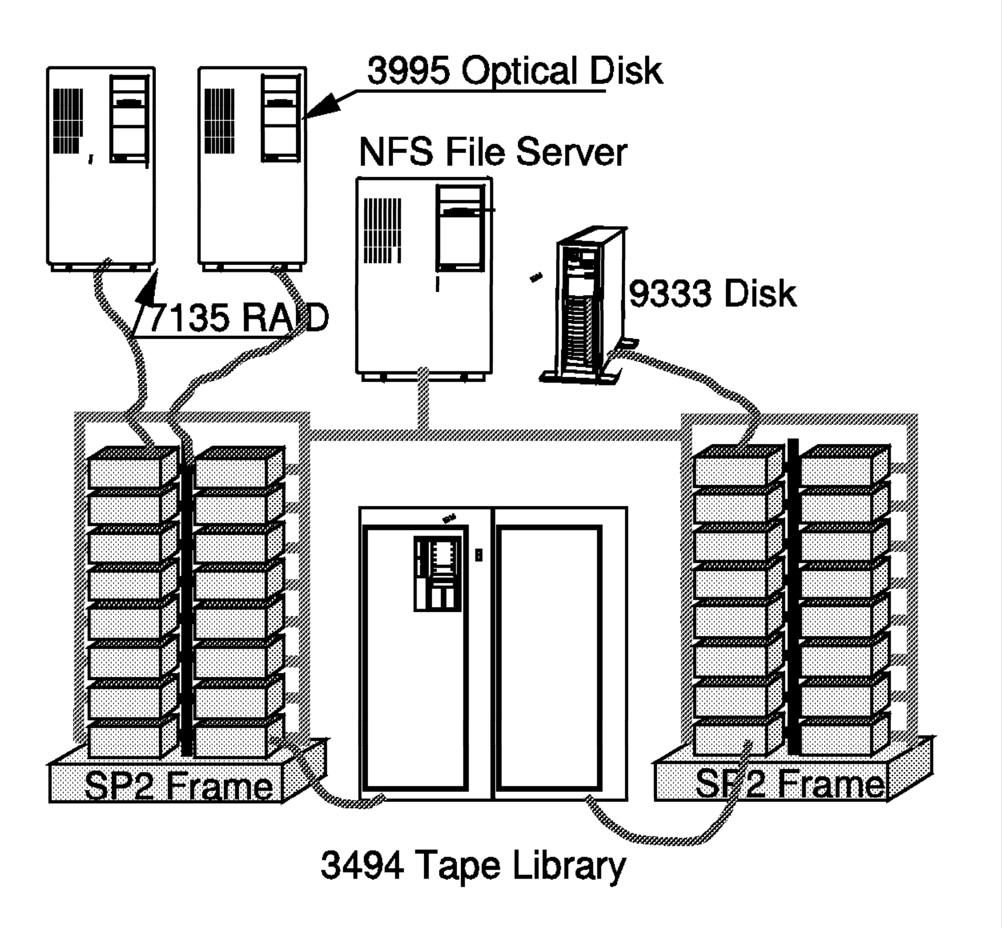


Figure 26. Hardware Configuration of Commercial Environment

  - IBM RISC System/6000 as NFS Dataserver

ADSM Server 2

- ADSM Server 2 runs on node16 (wide node).
- ADSM Server 2 serves all applications running in frame 2.
- ADSM Server 2 has access to:
  - IBM 9333 DASD
  - IBM 3494 Tape Library with 240 cartridges
- The IBM 3494 Tape Library is shared between ADSM/6000 server 1 and server 2.

The ADSM/6000 administrator in a commercial environment is likely to be the focal point for all backup/archive related operations on the system. The users themselves rely on implemented automated backup procedures. On the other hand, if they need to retrieve lost files, they want to perform this operation themselves or in some cases contact the ADSM/6000 administrator for help.

10.1 Business Application Backup/Archive Setup

The business applications in our scenario all run on node02. This node is a wide node equipped with enough data space to hold all the file systems locally.

Node02 File Systems

```
      local /(root)
      local /usr
      local /tmp
      local /var
      local /appl1
      local /appl2
```

Node02 requires installation of the ADSM/6000 client under the local file system /usr because it is considered a full AIX node. The data that should be backed up from this node is in file system /appl1 (business application 1) and /appl2 (business application 2), which are local file systems on this node for holding application related data. It is desirable to set up a management class on the ADSM/6000 server 1, which is controlling the data that comes from this client.

Management Class: business

```
Backup Copy Group
      Frequency                0
      Versions Data Exists     4
      Versions Data Deleted    4
      Retain Extra Versions    365
      Retain Only Version      999
      Serialization            shrstatic
      Mode                     Modified
      Destination              RAIDPOOL

no archive Copy Group
```

Notes:

- The destination of this data is the IBM 7135 RAIDiant Array because there is a high availability requirement for it.
- There will be four versions of a file maintained even if the file is deleted (on purpose or accidentally).
- Data will expire after a year, except if it is the only version of a file that remains. In that case, it will be kept for 999 days.
- Data will only be backed up if the file is not in use while the backup process is in operation.

The dsm.sys and the dsm.opt files are set as in the following example:

```
dsm.sys:
        server                 adsm_serv_1
        commethod              tcpip
        tcpserveraddress       node00.wtscpok.ibm.com
        tcpport                1500
        passwordaccess         generate
        inclexcl               /var/adsm/inclexcl.list

dsm.opt:
        domain  /appl1 /appl2

/var/adsm/inclexcl.list:
        include /appl1/.../*  business
        include /appl2/.../*  business
```

Note the include/exclude list used here. The second parameter in this list (in our case “business”) is the mechanism to direct the files into the “business” management class with a destination of the IBM 7135 RAIDiant Array.

The ADSM/6000 administrator will have to set up a schedule for this node which backs up all data twice a day (see definition of ADSM Server 1 in 10.5, “ADSM Server 1 Setup” on page 80). If additional incremental backups are desired, the following command executed on node02 from the root user is appropriate:

```
dsmc incremental
```

10.2 CAD Application Backup/Archive Setup

The CAD design application runs on node04 thru node15. All these nodes have some local file systems, such as /(root), /var, and /tmp, but the /usr file system is shared from node00; and the /CAD file system, where the data resides, is shared from the external NFS file server.

Node04 to Node15

```
local      /(root)
shared     /usr     from Node00
local      /tmp
local      /var
shared     /CAD     from the external NFS file server
local      /work
```

Each of these nodes needs to run the ADSM/6000 client. However, since they all share the /usr file system from node00, the ADSM/6000 client must only be installed on node00. All nodes can execute the ADSM/6000 client code. Some minor local customization should be done through symbolic links from /usr/lpp/adsm/bin to /var/adsm, such as described in 6.2.2, “Installation for /usr Clients” on page 39.

The data that we expect to backup from these clients actually resides on the IBM RISC System/6000 external file server. Only for the time of active design work will the data reside on the node itself. The approach we will try to implement is that the regular backup work is done from the external NFS file server. This has the advantage that the data does not have to travel twice across the network, once from the IBM RISC System/6000 external file server to the CAD node and then from the CAD node to the ADSM/6000 server. But instead, the external NFS file server sends the data directly to the ADSM server. This means that the ADSM AIX client has to be installed on the external NFS file server.

On the other hand, we want to provide a method for the CAD application designer so he can archive a finished design work and attach a small description, which allows him to easily find the file at a later time. Let us assume that the data file resides on the node in the local file system /work while the designer is working with the file. The management class on the ADSM/6000 server 1 should be set up as follows:

Management Class: cadarchive

```
Backup Copy Group              does not exist

Archive Copy Group
      Frequency                CMD
      Retain Version           nolimit
      Serialization            shrstatic
      Mode                     absolut
      Destination              3995POOL
```

Notes:

- The management class "cadarchive" does not have a backup copy group. This means that node04 to node15 workstations are limited to do archive operations only. See also the definition of a separate policy domain, called "CAD," on the ADSM/6000 Server 1 for this purpose.
- The destination of this data is the IBM 3995 Optical Library Dataserver because the optical disk is best suited to keep the large data files for a long period.
- The archived data has no expiration date.
- Only files that are not modified during the archive are backed up. If the file is open during the first attempt, the archive operation is repeated as many as four times.

The automatic backup setup for the CAD design data residing on the IBM RISC System/6000 external file server is done from the external file server itself. On the external file server an automatic backup schedule is defined that will back up this data. The management class used for this purpose is the following:

Management Class: cadbackup

```
Backup Copy Group
      Frequency                0
      Versions Data Exists     4
      Versions Data Deleted    1
      Retain Extra Versions    30
      Retain Only Version      365
      Serialization            shrstatic
      Mode                     Modified
      Destination              BACKUPPOOL

no archive Copy Group
```

Note:

1. The destination of this data is the internal DASD of ADSM server 1, which guarantees a fast retrieve.
2. The ADSM/6000 server will keep four versions of a file.
3. A file must not be modified during backup.

The dsm.sys and the dsm.opt files on the external NFS file server should have at least the following lines:

```
dsm.sys:
       virtualmountpoint      /tmp/CAD

dsm.opt:
       domain  /tmp/CAD

/var/adsm/inclexcl.list:
       include /tmp/CAD/.../*  cadbackup
```

The use of the “virtualmountpoint” parameter reduces the number of subtrees that have to be searched by the ADSM/6000 client backup process. In this example, only files located in /tmp/CAD would be included in an incremental backup. Note the include/exclude list used here to control the management class. The second parameter in this list (in our case “cadbackup”) is the mechanism to direct the files into the “cadbackup” management class with a destination of the IBM 7135 RAIDiant Array.

10.3 Publishing Application Backup/Archive Setup

Users of the publishing application are scheduled to one of the nodes 18 thru 21, where the publishing application is made available from /usr of node16. The result of their editing work is stored in /publish, which resides on the external NFS file server.

Node18 thru Node21

```
local   /(root)
shared  /usr    from node 16
local   /tmp
local   /var
shared  /publish from external NFS file server
```

All of the backup and restore operations are performed by an administrator who will always execute this work from the external NFS file server. The ADSM/6000 server 2 has a management class with the following information.

Management Class: pubbackup

```
Backup Copy Group
      Frequency                0
      Versions Data Exists     4
      Versions Data Deleted    1
      Retain Extra Versions    30
      Retain Only Version      999
      Serialization            shrdynamic
      Mode                     Modified
      Destination              9333POOL

no archive Copy Group
```

Notes:

- The destination of this data is the IBM 9333 DASD because there is a high demand to quickly restore a file.
- There will be four versions of a file maintained.
- Data will expire after 30 days, except if it is the only version of a file that remains. In that case it will be kept for 999 days.
- The file will be backed up, even if it is modified during the backup.

10.4 Office Application Backup/Archive Setup

The office application is accessible from node22 thru node31. Users of the office application are scheduled to one of these nodes automatically, and for them it is transparent to which node they are actually assigned. The /home filesystem and the /usr filesystem of these office users is NFS mounted from node16.

Node22 to Node31

```
local  /(rrot)
shared /usr   from node16
local  /tmp
local  /var
shared /home  from node16
```

The data that we want to backup from these nodes is physically located at the NFS /home server in node16. This means that node16 must run an ADSM/6000 client and backup all this data in /home/users. At the ADSM/6000 server 2 we will setup a management class that controls backup parameters, which includes the following:

Management Class: office

```
Backup Copy Group
      Frequency                 0
      Versions Data Exists      4
      Versions Data Deleted     4
      Retain Extra Versions     365
      Retain Only Version       999
      Serialization             static
      Mode                      Modified
      Destination               RAIDPOOL

no archive Copy Group
```

We also want these office users to be able to restore the files themselves. To perform this operation, nodes22 to node31 must be authorized to access the data saved by node16. You must therefore run the following sequence of ADSM/6000 commands on node16 to provide the correct permissions.

```
dsmc set access backup ″/home/usera/*″ ″*″ node22
dsmc set access backup ″/home/usera/*″ ″*″ node23
  ⋮
dsmc set access backup ″/home/usera/*″ ″*″ node31
```

Repeat this sequence once for every user on the system.

The ADSM/6000 client must also be executable from the office application nodes. This is achieved easily because all office nodes share the /usr filesystem from node16 where the ADSM/6000 client is installed. Some local customization must be performed, as described in Section 7, “ADSM Basic Customization” on page 41.

In addition to the previous service, we want to allow the office users to archive files from their /home directory. To do this, we must create another management class on the ADSM/6000 server which only allows use of the archive function of ADSM/6000 and which will send all archived files directly to the IBM 3494 Tape Library.

Management Class: officearchive

```
No  Backup Copy Group

Archive Copy Group
      Frequency                CMD
      Retain Versions          365
      Serialization            static
      Mode                     absolut
      Destination              3494POOL
```

Notes:

- Data will be sent directly to the IBM 3494 Tape Library.
- Data will expire after a year.

10.5 ADSM Server 1 Setup

ADSM/6000 Server 1 runs on node00. This node has access to several I/O devices, including 60GB of RAID disk, 40GB of optical storage and access to the IBM 3494 Tape Library.

The ADSM/6000 server 1 has four storage pools:

3995POOL on the IBM 3995 Optical Library Dataserver
This pool is used purely for archiving the CAD images. The lifetime of the data is unlimited.

RAIDPOOL on the IBM 7135 RAIDiant Array
This pool will be used for all backup data that requires a fast retrieve and high availability.

3494POOL on the IBM 3494 Tape Library
The 3494 pool will hold all long term data and eventually pick up any overflows from the other two pools.

BACKUPPOOL on the internal DASD
The BACKUPPOOL on the internal DASD holds short term data with fast retrieve capability.

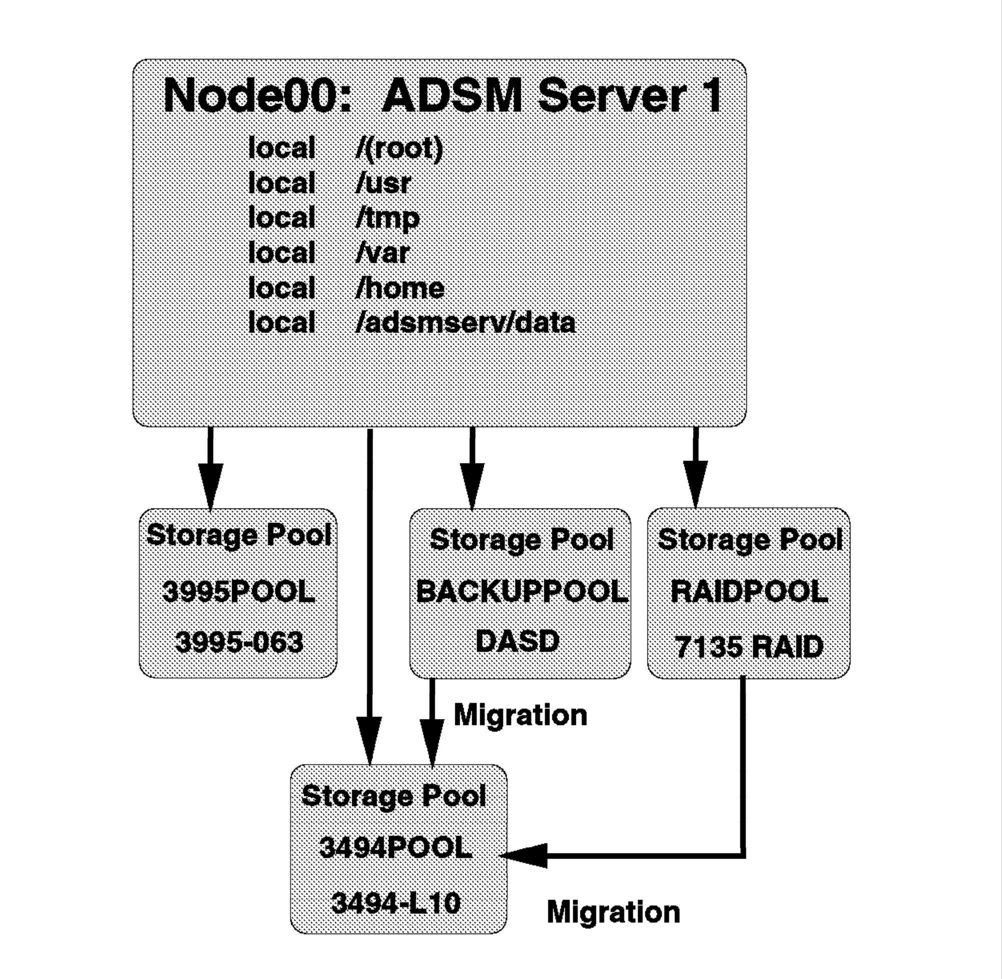


Figure 27. ADSM Server 1 Setup on Frame 1 for Commercial Scenario

ADSM/6000 Server 1 will have two policy domains. One for node00 and node02, the other for node04 to node15.

The policy domain *STANDARD* is established for node00, node02 and the external NFS file server. It controls the backup activity from these nodes. There is no archive operation supported because the policy domain does not provide an archive copygroup.

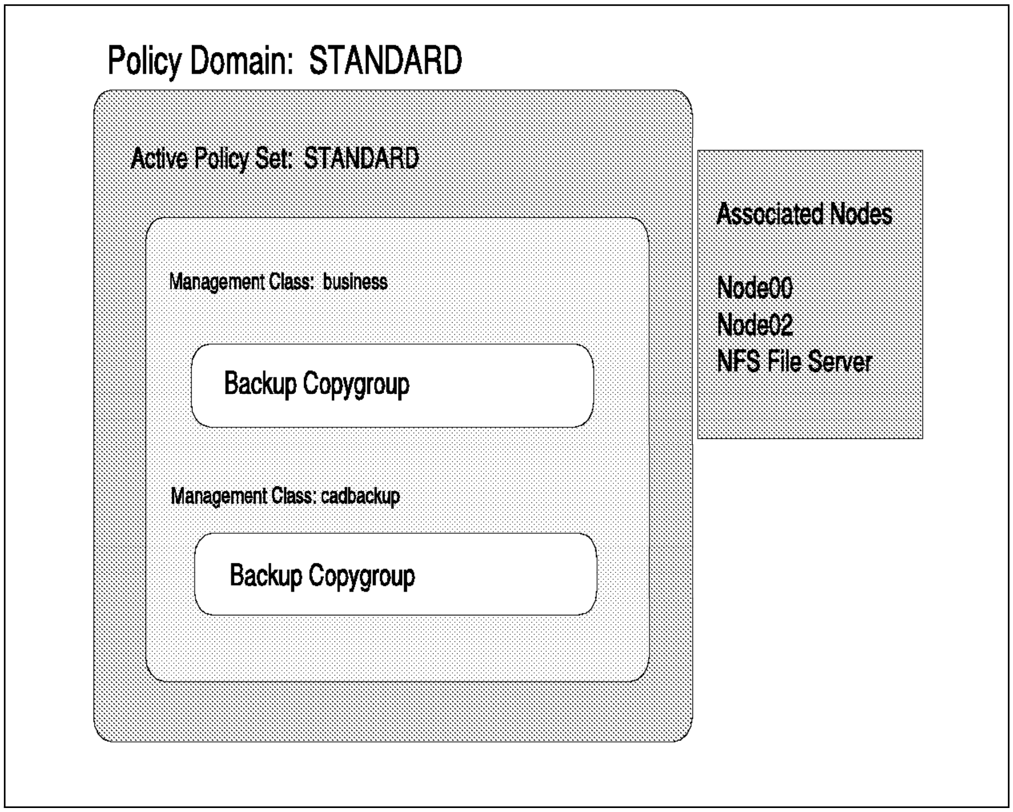


Figure 28. STANDARD Policy Domain

The policy domain *CAD* is created for node04 to node15. The users of the CAD design application use this policy domain to archive their design work on the IBM 3995 Optical Library Dataserver. Notice that the policy domain has no backup copygroup. This restricts the users of node04 to node15 to perform just archives but no backups.

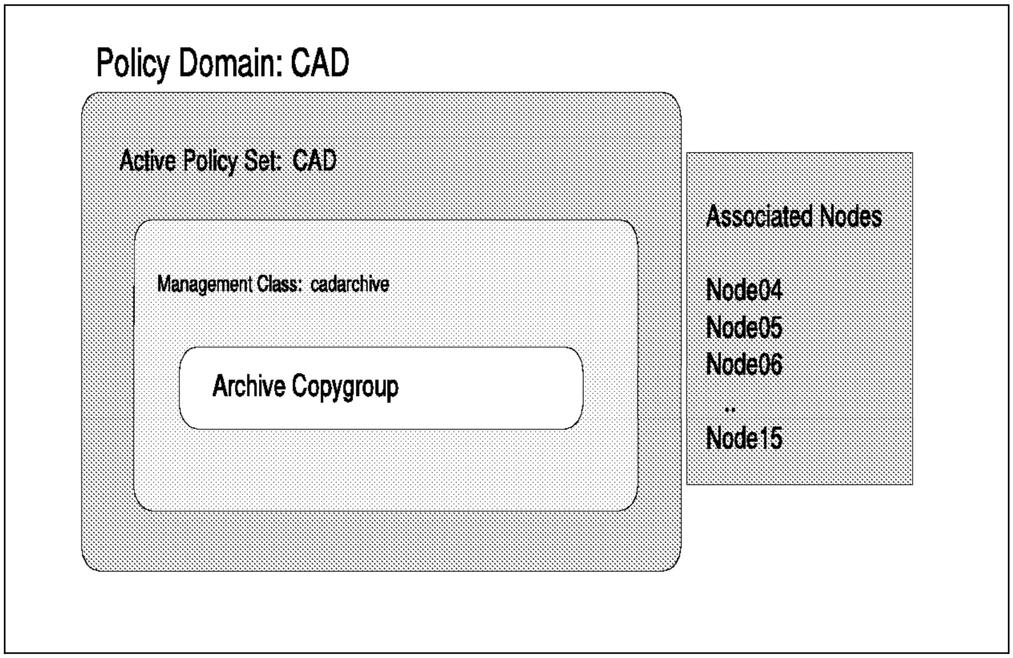


Figure 29. Policy Domain CAD

The following schedule is set up by the ADSM/6000 administrator and controls the backup schedule for business users. Recall that the data for these applications resides on node02. That is why we run this schedule with only one associated node, which is node02.

Schedule: BUSINESS

```
Policy Domain:      Standard
Action:             incremental
Start Date/Time     2:00 am  11/11/94
Duration:           2 hours
Period:             every 12 hours
Priority:           5
```

Notes:

- Associated node to the BUSINESS schedule is only node02.
- Schedule runs twice a day.

The schedule named CAD is supposed to back up the CAD design data, which resides on the external NFS file server.

Schedule CAD

```
Policy Domain:      Standard
Action:             incremental
Start Date/Time     2:00 am 11/11/94
Duration:           3 hours
Period:             1 per day
Priority:           5
```

Notes:

- Associated node to the CAD schedule is only the external NFS file server.
- The external NFS file server backs up the CAD design data once every day.

Table 2 on page 87 summarizes the classes of applications we have defined along with the policies and schedules we have set up on the server.

10.6 ADSM Server 2 Setup

ADSM/6000 server 2 runs on node16. This node has access to 8GB of external disk on the IBM 9333 DASD, and to the IBM 3494 Tape Library.

The ADSM/6000 server 2 has two storage pools:

9333POOL on the IBM 9333 DASD
This pool is used for short term data, such as the data that results from the backup of daily scheduling.

3494POOL on the IBM 3494 Tape Library
This pool will act as the overflow area for all data that does not fit into the 9333POOL.

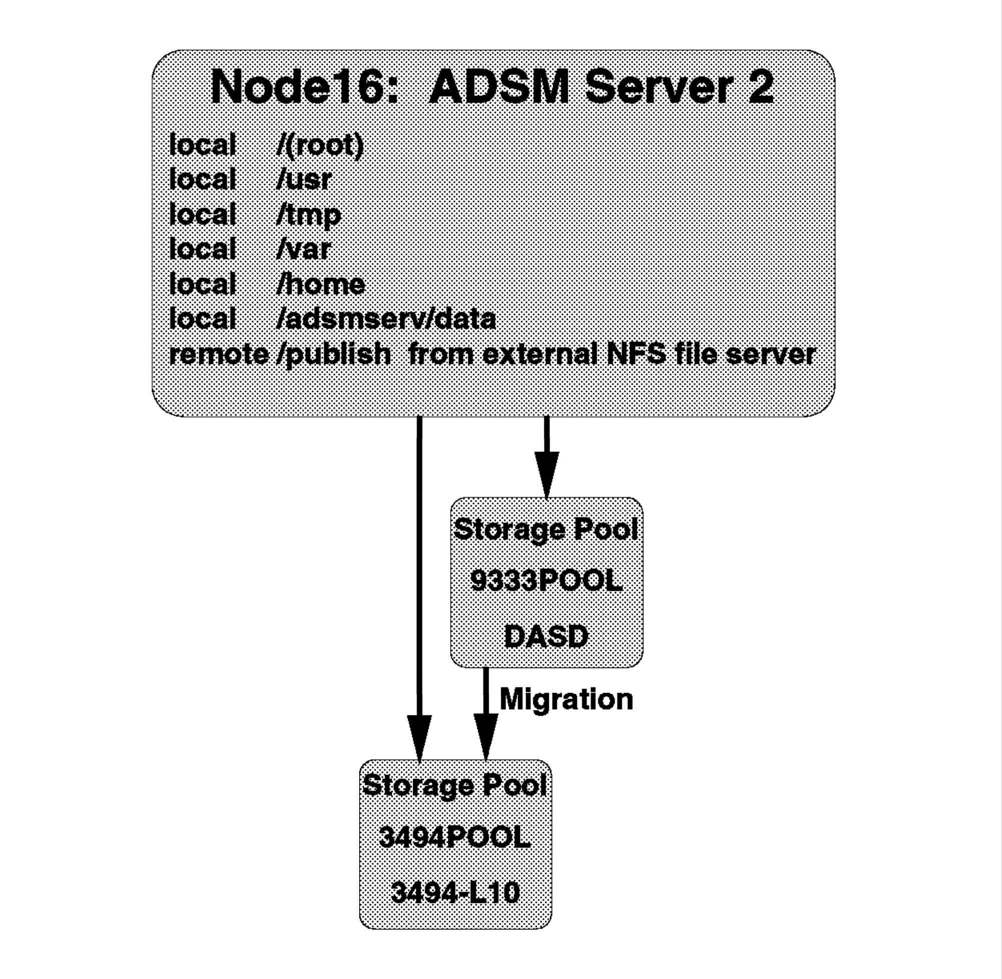


Figure 30. ADSM Server 2 Setup on Frame 2 for Commercial Scenario

The policy domain *STANDARD* is created to backup the files from the office application (management class: office) to allow users of the office application to archive files from their /home directories (management class: officearchive), and to back up files from the publishing application.

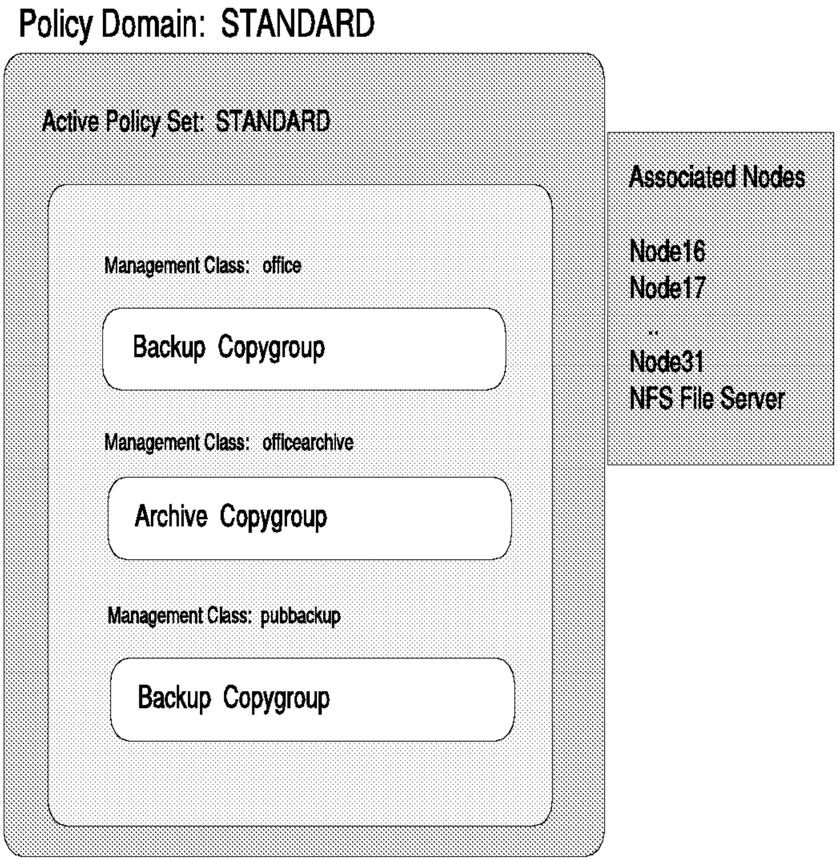


Figure 31. Policy Domain STANDARD

The schedule *office* is created to periodically back up the files from the office application.

Schedule: office

```
Policy Domain:      Standard
Action:             incremental
Start Date/Time     2:00 am  11/11/94
Duration:           2 hours
Period:             1 per day
Priority:           5
```

Notes:

- Associated node to the office schedule is only node16.

The schedule *publish* is created to periodically back up the files from the publishing application.

Schedule: publish

```
Policy Domain:      Standard
Action:             incremental
Start Date/Time     2:00 am 11/11/94
Duration:           3 hours
Period:             1 per day
Priority:           5
```

Notes:

- Associated node to the publish schedule is only the external NFS file server.

Table 2 on page 87 summarizes the classes of applications we have defined along with the policies and schedules we have set up on the server.

Table 2. Commercial Users Backup and Archive Strategy

ADSM Client	File System	Type of Applic.	Backup or Archive	Management Class	Policy Domain	ADSM Server	Schedule
Node02	/appl1, /appl2	business	backup	business 1	STANDARD	1	business, twice a day
Node04 thru Node15	/work	temp. CAD design data	archive	cadarchive 2	CAD	1	none
external NFS File Server	/CAD	CAD design data	backup	cadbackup 3	STANDARD	1	once a day
	/publish	documents	backup	pubbackup 4	STANDARD	2	publish, once a day
Node16, /usr server	/usr	home directories	backup	office 5	STANDARD	2	office, once a day

Note:

1 Management Class: business: Backup to RAIDPOOL (IBM 7135 RAIDiant Array).

2 Management Class: cadarchive: Archive to 3995POOL (IBM 3995 Optical Library Dataserver).

3 Management Class: cadbackup: Backup to BACKUPPOOL (internal DASD).

4 Management Class: pubbackup: Backup to 9333POOL (IBM 9333 DASD).

5 Management Class: office: Backup to RAIDPOOL (IBM 7135 RAIDiant Array).

Chapter 4. Administration and Operations

The specific architecture of POWERparallel systems implies particular rules in the administration and operation of ADSM/6000 on IBM POWERparallel Systems SP2. This chapter describes several topics that need particular attention, and suggests solutions to help ADSM administrators. These topics are included in following sections:

- System management
- Multiple ADSM/6000 servers
- Performance

Section 11. System Management

This section describes the issues we have encountered related to ADSM management on SP2. The problems are related to the following issues:

- Uniform user definition across the nodes
- Centralized management of client′s error and schedule logs
- Using the IBM 3490E Model E11 Tape Unit
- Keeping the scheduler alive and updating the scheduler with the latest contents of dsm.sys and dsm.opt
- Using ADSM macros
- Using the ADSM/6000 accounting system
- Handling the activity log
- Handling the event log

11.1 SP2 Users Definition to ADSM

ADSM users can access only the data that they own or they have been explicitly authorized to access. An ADSM user owns the data that he has stored. That means that he can retrieve the data from the SP2 node where he saved the data by using the same user ID. However, SP2 users are considered to be equivalent on all the nodes, and they expect to be able to retrieve their data from any node. This can be accomplished by the administrator explicitly authorizing the users to access data from any node in the SP2 system. However, granting access from all nodes with a command like the following can be dangerous:

```
dsmc set access backup /home/user01/* * user01
```

In fact, it allows a user named user01 running on any node of the network (not only SP2 nodes) to access the data. For each ADSM/6000 client, the ADSM administrator should run the authorization command separately on the NFS server whose data the user wants to access. This would be achieved with a series of commands such as:

```
dsmc set access backup /home/user01/* node00 user01
dsmc set access backup /home/user01/* node01 user01
dsmc set access backup /home/user01/* node02 user01
   ⋮
dsmc set access backup /home/user02/* node00 user02
dsmc set access backup /home/user02/* node01 user02
dsmc set access backup /home/user02/* node02 user02
   ⋮
```

That operation is time consuming and error prone, so the ADSM administrator can use a short shell script, like the one listed in Appendix C, “SP2 Users Authorization” on page 155 to automatically extract the list of SP2 nodes and run the authorization command on each node. The shell optionally takes a list of nodes on which to run the command. That feature is useful in case the NFS server serves only a subset of the nodes and running the command on the other nodes would either fail or be useless.

11.2 Centralized Management of Logs

In the day-to-day operation, ADSM/6000 produces several log files. Log files are written by the scheduler process on the client side, and log files are also created by the ADSM/6000 client in case of errors. For the administrator of the SP2, who views the SP2 as a single system, it is very disturbing to look for these log files on each node. We would like to show a way of collecting the log files from all nodes in a single directory, which could be on the control workstation or on the node that runs the ADSM/6000 server. This is a proposal, and the code examples shown in this section are not part of IBM Licensed Products.

The log files that we care about are:

dsmerror.log

The error log contains information about any errors that occur during processing of the ADSM/6000 client process. The dsmerror.log file is created by default in the current directory. If the environment variable DSM_LOG specifies a directory, the dsmerror.log is created in that directory, but the process must have write permission to that directory. If the process has no write permission, it will default to the current directory.

dsmsched.log

The schedule log contains information about the scheduling process on the ADSM/6000 client side. In the log you can find detailed information about when the scheduler started and what files actually have been executed by the scheduler. The default place for the dsmsched.log is the current directory, and it has a file name of dsmsched.log. You can define the full pathname of the scheduler log file by setting the option SCHEDLOGNAME in the dsm.sys customization file.

We recommend using the control instruments provided by ADSM/6000 to define the output files of these two log files. Let us look at the example showing how we solved this problem in our lab environment.

1. Define the directory for the dsmerror.logs in each node to be the /var/adsm/logs directory. To do this, we set the DSM_LOG environment variable. A good place to do this is to add it to the /etc/environment file. The *dsh* command of the SP2 allows you to add this easily to all nodes at the same time. If the WCOLL environment variable is set to point to a file that contains all nodes, you need to enter the following command:

   ```
   # dsh ′echo ″DSM_LOG=/var/adsm/logs″ >> /etc/environment′
   ```

   All users logging in to the system and all processes (foreground or background) use the /etc/environment file. We also provided write permission for everybody so that the ADSM/6000 processes could create and write to /var/adsm/logs/dsmerror.log:

   ```
   # dsh mkdir /var/adsm/logs
   # dsh chmod a+w /var/adsm/logs
   ```

2. Define the complete pathname of the dsmsched.log file for each node to be /var/adsm/logs/dsmsched.log. This is accomplished by setting the option SCHEDLOGNAME in the dsm.sys file, such as:

```
# pg /var/adsm/dsm.sys
   ⋮
SCHEDLOGNAME    /var/adsm/logs/dsmsched.log
   ⋮
```

What we have achieved so far is that each node will have only one place where all the log files are collected. This is the directory /var/adsm/logs on each node of the SP2 where the ADSM/6000 clients run.

In the next step we show how to send all the log files to a centrally managed directory. In our lab environment we selected the /var/adsm/logs directory on node02, that is the one that runs the ADSM/6000 server, to collect all logs from all nodes. The nodes will execute a shell script daily and send their log files to the central directory. To distinguish the logs once they are in the central directory, we append the date and the hostname to the log files.

An example of the shell script, which is invoked thru cron every day on each node, is outlined here:

```
#!/bin/ksh
#******************************************************************
# send all log files (dsmerror.log and dsmsched.log) to
# the central log directory located on node02
# P.Koepp, 11/24/94,
#******************************************************************
logdir=/var/adsm/logs
if [[ -s dsm.errolog ]]; then
 rcp $logdir/dsmerror.log node02:$logdir/dsmerror.log.′date +%m%d′.′hostname ′
 if [[ ″$?″ = ″0″ ]]; then
      rm $logdir/dsmerror.log
     fi
fi
if [[ -s dsmsched.og ]]; then
 rcp $logdir/dsmsched.log node02:$logdir/dsmsched.log.′date +%m%d′.′hostname ′
 if [[ ″$?″ = ″0″ ]]; then
      rm $logdir/dsmsched.log
     fi
fi
```

If this script runs daily on every node, we will get a collection of the dsmerror and dsmsched logs for each day in the central directory, which will look something like the following:

```
ls /var/adsm/logs
   ⋮
dsmerror.log.1111.node13
dsmerror.log.1111.node14
dsmerror.log.1111.node15
dsmerror.log.1111.node16
dsmerror.log.1112.node13
dsmerror.log.1112.node14
dsmerror.log.1112.node15
dsmerror.log.1112.node16
dsmerror.log.1113.node13
dsmerror.log.1113.node14
dsmerror.log.1113.node15
dsmerror.log.1113.node16
   ⋮
```

And the schedule logs will be collected as:

```
ls /var/adsm/logs
  ⋮
dsmsched.log.1111.node13
dsmsched.log.1111.node14
dsmsched.log.1111.node15
dsmsched.log.1111.node16
dsmsched.log.1112.node13
dsmsched.log.1112.node14
dsmsched.log.1112.node15
dsmsched.log.1112.node16
dsmsched.log.1113.node13
dsmsched.log.1113.node14
dsmsched.log.1113.node15
dsmsched.log.1113.node16
  ⋮
```

Last but not least, since these logs will add several new files every day to the /var/adsm/logs directory on node02, we must set up some kind of housekeeping routine to remove log files once they age. In our example, we delete all log files if they get older than three days. This is done by the housekeeping shell script run daily on node02 with the central directory for log files. The shell is outlined here:

```
#!/bin/ksh
#*****************************************************************
# perform housekeeping on the log files
# remove dsmerror logs and dsmsched logs once they are older
# than three days.
# P.Koepp, 11/24/94,
#*****************************************************************
logdir=/var/adsm/logs
find $logdir  -mtime +3 -name ’*dsmerror*’ -exec rm {} \;
find $logdir  -mtime +3 -name ’*dsmsched*’ -exec rm {} \;
```

11.3 Using the IBM 3490E Model E11

The IBM 3490E Model E11 is a high performance buffered magnetic tape storage subsystem with a control unit and one tape drive. The tape unit has a cartridge stack loader, which is an automatic cartridge accessing device that houses up to seven cartridges. The cartridge stack loader (hereafter called CSL) operates in three different modes, which are selectable from the front panel of the device:

- Manual mode
- Automatic mode
- System mode

All three modes work with ADSM/6000. There are certain benefits that ADSM/6000 can take advantage of when using either the automatic or the system mode. However, be alerted to the fact that this device is *not* usable as a tape library. This means ADSM/6000 has no knowledge of the cartridge stack loader as being a device (there is no /dev/lb0), and ADSM/6000 does not know which cartridges are actually loaded in the CSL. You, as the operator of the IBM 3490E Model E11, can only preload the CSL with those cartridges which are most likely to be requested from ADSM/6000. It is best to preload the CSL with scratch tapes.

Before using the IBM 3490E Model E11 together with ADSM/6000, you must install the device driver that ships with the IBM 3490E Model E11 itself. For many other tape drives, ADSM/6000 ships special device drivers, not for the IBM 3490E Model E11.

Note

When installing IBM 3490E Model E11 device driver, look for A3490E as the keyword when selecting the correct driver.

As mentioned before, ADSM/6000 works in any of the three modes: manual, automatic, or system. We experienced the best use of the IBM 3490E Model E11 with ADSM/6000 in the automatic mode with further selection of configuration modes of STOP and SIX+1.

What is AUTOMATIC, STOP, and SIX+1?

When the CSL is running in automatic mode, cartridges are automatically loaded from the CSL input stack following a system issued unload command. Cartridges are selected sequentially, from top to bottom. The STOP selection ensures that after all of the selectable cartridges in the input stack have been accessed, the CSL stops. The message End Stk is then displayed at the operator panel. The SIX+1 mode treats the first shelf in the CSL as a priority shelf. Any time a cartridge is loaded in the priority shelf, it will be processed on a priority basis. That is, when the operator opens the CSL door, loads a cartridge in the priority slot and presses the START button, the CSL enters the priority processing, and loads the priority cartridge after unloading the currently mounted cartridge.

Why is AUTOMATIC, STOP, and SIX+1 useful with ADSM/6000?

We assume that ADSM/6000 is used more for backup/archive than for restore/retrieve operations. That is, there will be many requests for scratch tapes and just occasionally (when a user requests a restore) a request for a specific tape will come up. That might not be the case, but if it is, the operator would preload the six lower shelves with scratch volumes and leave the top shelf empty. This means, whenever ADSM/6000 requests a scratch tape, it will find one from the lower six shelves until the last scratch tape is full. The IBM 3490E Model E11 would then display the End Stk message, which will alert the operator to remove the tapes and preload again six new scratch tapes. The ADSM/6000 storage pool for the tapes should be defined with a “maxscratch=number” parameter so that ADSM/6000 will take any scratch volume which is labeled properly. On the other hand, if ADSM/6000 requests a specific tape, the operator can satisfy that request by placing the correct cartridge into the priority shelf and pressing the START button. When ADSM/6000 has finished using the priority tape, it will put it back into the priority shelf, and the operator should remove the volume and store it.

How would an operator set up the system for the night?

Let us assume that ADSM/6000 schedules backup during the night, and that the operator wants to set up the IBM 3490E Model E11 so that scratch tapes are automatically loaded. ADSM/6000 will only request scratch tapes for further backup if all other tapes are full; and if any tape has a status of “Filling,” it will be the first one that will be requested. The idea is to set up the tape drive in such a way that when it requests the first tape, that is the one with status “Filling,” the CSL will provide this tape first and thereafter load scratch tapes. Follow these steps to set up the drive for the nightly incremental backup:

1. Determine which tape has a status of “Filling”:

   ```
   adsm> query volume
   ```

   Write down the label of the tape.

2. Unmount any tape that may be in the drive:

   ```
   adsm> dismount volume <label>
   ```

3. Remove all tapes from the CSL:

   ```
   open Door
   remove all tapes from CSL
   ```

4. Now place the tape volume, which has the status of “Filling,” into the second slot from the top, and place scratch tapes into the remaining five slots downwards.

   ```
   place the partially used volume (status 'Filling')
   place scratch volumes in remaining slots
   close door
   press START
   ```

   This ensures that the first volume which ADSM/6000 will request is already loaded in the drive when the backup starts. ADSM/6000 will write to it until it is full and request a scratch volume, which will then be loaded automatically. If the six tapes provide enough space for the nightly backup, it should complete automatically.

How to label new scratch tapes

The IBM 3490E Model E11 offers a single command that can be used to label all the tapes that are in the CSL. This way is more efficient for larger numbers of tapes, since you do not need to mount each one separately.

1. Place new scratch tapes into the bottom six slots of the CSL:

   ```
   open door
   put new scratch tapes into CSL (bottom six slots)
   close door
   press START
   ```

   The first scratch tape should be loaded automatically.

2. Label the scratch tapes:

   ```
   dsmlabel -drive=/dev/rmt0 -overwrite < ./labelfile
   ```

   where the file “labelfile” holds the labels, line by line, such as in:

```
tape00
tape01
tape03
tape04
  ⋮
```

The *-overwrite* parameter should only be used if you intend to change a label on a cartridges; otherwise pre-labeled cartridges would be refused.

11.4 Keeping the Scheduler Alive

With central scheduling, an administrator can set up schedules to run incremental or absolute backups on a regular basis and can schedule selective backups and archives. Central scheduling is a cooperative effort between the server definitions and the client scheduler code to perform the operation automatically.

For the successful operation of the scheduled backups, it is important that the client scheduler process is continuously up and running. The client process of the central scheduler will periodically request information about any schedules that have been set up for him. How often the client will contact the ADSM/6000 server and check for scheduled work can be defined with the QUERYSCHEDPERIOD option in the dsm.sys file (the default is every 12 hours).

It is most convenient to run the client scheduler in the background and write the output to a file that is selectable with the SCHEDLOGNAME option. So you can just start the client scheduler once when the system is IPLed. For example, put the following line into the /etc/inittab:

```
adsmsched:2:respawn:/usr/lpp/adsm/bin/dsmc scheduler
```

However, be aware that the client scheduler will not have knowledge of any changes made to the dsm.sys or dsm.opt files until it is killed and restarted. For example, changes made to the INCLUDE/EXCLUDE list while the scheduler is running will not be picked up by the *dsmc* process. An alternative is to use the following shell script to check if the scheduler is still alive, eventually kill it and restart it. If you run this script once a day from *cron*, you are sure that the scheduler is alive and that it also picks up the latest changes you may have made to the dsm.sys and dsm.opt files.

```
#!/bin/ksh
##############################################################
# check the ADSM scheduler
# If scheduler is alive, then kill the scheduler and restart,
# so it picks up new values from dsm.sys and dsm.opt
# Otherwise just start the scheduler
# P. Koepp,  11/26/94
##############################################################
sched=′ps -eaf | grep -c dsmc′
if [[ $sched -gt 1 ]]; then
   procnum=′ps -eaf | grep dsmc | grep -v grep | cut -c10-14′
   kill -9 $procnum
fi
/usr/lpp/adsm/bin/dsmc sched &
```

11.5 Using ADSM Macros

ADSM/6000 provides a function that allows writing macros to collectively issue a series of administrator commands. A macro is a standard ASCII file that contains administrative commands. It can only be issued from the administrative client in a batch or interactive mode, not from the ADSM/6000 server console. Backup/archive commands are not able to be issued from macros.

Macros are especially useful if you have a series of commands that have to be executed repeatedly. In such a case you would edit all required ADSM commands into a file and store this file at any convenient location on the ADSM client system from which you intend to start the administrative client program. For example, if you have a need to lock a group of ADSM clients from time to time, you can write two macros for this purpose. One to lock the group and another to unlock the group.

Let us look at an easy example where we lock and unlock a group of users. Edit a file with the name *lock* with the following lines:

```
lock node user1
lock node user2
lock node user3
lock node user4
lock node user5
```

When you need to lock this group of users from any ADSM services, call this macro as:

```
adsm> macro lock
```

It is easy to understand how the *unlock* macro would work.

Macros can be nested as well. This means that you can call a macro from within another macro, up to 10 levels deep.

A nice feature of the macro utility is the *commit* and *rollback* function. This allows you to group commands in a macro and complete the processing with a single commit statement past the group of commands. If an error occurs during the execution of any command in the macro, the server terminates processing and rolls back any changes caused by all previous commands.

Another example of using the macros is when you have a set of commands that you want to apply to existing policies and change these policies quickly and consistently in one step. Look at the macro called *chday*, which allows you to change the day of the backup for all schedules.

```
update schedule aixdomain schedname1 -
              perunits=day -
              dayofweek=%1
update schedule os2domain schedname2
              perunits=day -
              dayofweek=%1
Commit
```

Use this macro with the following command:

```
adsm> macro chday sunday
```

The string *sunday* is a parameter, and the term “%1” in the macro is substituted with this string.

11.6 ADSM/6000 Accounting

ADSM/6000 uses its own accounting processing. This should not be misunderstood to be the accounting system that is built into the AIX operating system. The ADSM/6000 accounting component comes with ADSM/6000 and only works for ADSM/6000. By default it is off, and if you want to use ADSM/6000 accounting, use the following command to turn it on:

```
adsm> set accounting on
```

The accounting system will write records, one for each session, to a file called *dsmacct.log* in the directory from where the server is started. Since the server is usually started from the /usr/lpp/adsmserv/bin directory, it may not be convenient to produce the accounting records in the /usr file system, which should be more read oriented. Especially on the IBM POWERparallel Systems SP2, when you use a shared /usr file system, it is not appropriate to write to /usr. Therefore, create a directory on the local file system, for example /var/adsmserv, and place the accounting record there by installing a symbolic link such as:

```
/usr/lpp/adsmserv/bin/dsmacct.log ---> /var/adsmserv/dsmacct.log
```

This is accomplished with the following commands:

```
mv /usr/lpp/adsmserv/bin/dsmacct.log /var/adsmserv/dsmacct.log
ln -s /var/adsmserv/dsmacct.log /usr/lpp/adsmserv/bin/dsmacct.log
```

The file is written as text records and can be viewed with an editor. There are 24 fields, separated by commas, and a record is terminated with a newline character. The exact information in each field is listed in the *ADSM/6000 Administrator′s Guide*.

Currently there are no tools shipped with ADSM/6000 that allow you to work with these accounting records. However, it is possible to read the records into a Lotus 123 spreadsheet program.

11.7 Handling the ADSM Activity Log

ADSM provides an activity log to help you track server activity and monitor the system. Examples of the kind of messages you would find in the activity log are:

- When a client session begins and ends
- When migration begins and ends

- Information about backup operations
- Information about export/import processing
- Information about tape handling

The activity log is actually inside the ADSM database, and when the size of the ADSM database is determined, it must reserve some extra space for the activity log. It is recommended to allocate at least 1MB of additional space in the ADSM database for the activity log. The size of the activity log is of course dependent on the activity in the ADSM/6000 server itself but is also dependent on the lifetime of each individual message. The default lifetime of activity log messages is one day, but this parameter is adjustable by administrator commands.

If you need to see the current setting of the activity log retention period, query the status of the server with the following command:

```
adsm> query status
DSM Server for AIX-RS/6000 - Version 1, Release 2, Level 0.4/1.4


                      Server Name: SP2SW02
    Server Installation Date/Time: 10/28/1994 13:55:39
         Server Restart Date/Time: 11/17/1994 16:16:46
                   Authentication: On
       Password Expiration Period: 1 Day(s)
                     Registration: Closed
                     Availability: Enabled
                       Accounting: On
    Activity Log Retention Period: 1 Day(s)
   Total Storage Licensed for Use: 50 G
            Licensed Client Nodes: 25
             License Audit Period: 30 Day(s)
               Last License Audit: 11/29/1994 15:40:02
                Central Scheduler: Active
                 Maximum Sessions: 25
       Maximum Scheduled Sessions: 12
    Event Record Retention Period: 10 Day(s)
Schedule Randomization Percentage: 50
            Query Schedule Period: Client′s Choice
          Maximum Command Retries: Client′s Choice
                     Retry Period: Client′s Choice
                 Scheduling Modes: Any
```

If you want to change the retention period of the activity log, use the following command to keep it, for example, for 10 days:

```
adsm> set actlogretention 10
```

How would you use this information? One way is simply to list the information from the activity log. However, the list is not very readable if everything is dumped to the screen. It is better to use some filtering techniques to see just the information that you are interested in. For example, if only those messages that hold the string “error” should be displayed, use the following command:

```
adsm> q actlog begindate=11/29/1994 search=″error″

Date/Time            Message
-------------------- ----------------------------------------------------------
11/29/1994 15:30:09  ANR1411W Access mode for volume TAPE09 now set to
                      ″read-only″ due to write error.
11/29/1994 15:30:09  ANR0523W Transaction failed for session 346 - error on
                      output storage device.
```

Or if the message number is known that we want to search for, we could use the following command:

```
adsm> q actlog begindate=11/29/1994 msgno=8303

Date/Time            Message
-------------------- ----------------------------------------------------------
11/29/1994 15:32:52  ANR8303E I/O error on drive 3490DRV (/dev/rmt0)
                      (OP=LOADVOLUME, SENSE=40.40.24.00).
11/29/1994 15:34:05  ANR8303E I/O error on drive 3490DRV (/dev/rmt0) (OP=READ,
                      SENSE=00.00.00.00).
```

This is useful if a particular error shows up once in a while, and the message number is known in advance so that we can quickly check if the error has occurred again during the past few days.

11.8 ADSM Scheduler Event Log

Event records are ADSM database records that describe the results of scheduled events, such as backup or archive operations. An event log entry is created whenever a scheduled command is started. The status of the command such as completed, failed, or missed is recorded in the event log. The event records are logged for a length of time that is specified by the EVENTRETENTION parameter in the ADSM server. The default is 10 days, but it can be changed to any other desirable number of days ranging from 0 to 9999. An example is shown here:

```
adsm> set eventretention 50
```

Consider extra space in the ADSM database if you specify the retention period to be longer than the default 10 days.

For the ADSM administrator it is very useful to know how to filter the event log and extract the information needed. Normally, the event log is too long to be displayed all at one time. There are ways to filter the result of an event log query by specifying ADSM domains, ADSM schedule names or client node names. Let us look at some examples. The following command displays a list of all events that have been scheduled for the current day.

```
adsm> q event * *
Scheduled Start: 12/01/1994 02:00:00
   Actual Start: 12/01/1994 03:07:38
  Schedule Name: HOMEDIR
      Node Name: SP2CW0
         Status: Completed

Scheduled Start: 12/01/1994 07:53:00
   Actual Start: 12/01/1994 07:53:41
  Schedule Name: COS
      Node Name: SP2SW03
         Status: Failed
```

The next command will display all events within a certain time range, and a nice feature here is the *exceptionsonly=yes* parameter, which filters only those records that hold transactions that have failed or did not complete. This is a convenient way to check quickly if any errors have occurred during automatic scheduled backups or archives.

```
adsm>
q event * * begindate=11/29/1994 enddate=12/01/1994 exceptions=yes

Scheduled Start: 11/29/1994 02:30:00
   Actual Start:
  Schedule Name: APPLIDIR
      Node Name: SP2SW14.ITSC.POK.IBM.COM
         Status: Missed

Scheduled Start: 12/01/1994 07:53:00
   Actual Start: 12/01/1994 07:53:41
  Schedule Name: COS
      Node Name: SP2SW03
         Status: Failed
```

Since event records are automatically removed when the retention period has passed, there is no need to delete them manually. However, a reason to do it manually would be to release space from the ADSM database. Therefore, ADSM/6000 provides a command to delete event record entries from the database. The following command is an example of deleting all event records prior to the date 11/29/1994 and before the time 11:00 AM.

```
adsm> delete event 11/29/1994 11:00
```

Section 12. Multiple ADSM/6000 Servers

This section deals with multiple ADSM/6000 servers. Is explains briefly how to install multiple ADSM/6000 servers on a /usr server system, shows how to use them from an ADSM/6000 client and provides some planning help for multiple server environments.

12.1 Installing Multiple ADSM/6000 Servers

There are two ways of installing multiple ADSM/6000 servers on the IBM POWERparallel Systems SP2. One way is to install the server code on several nodes, which have a full AIX operation system (all file systems are local). The other method is to install the ADSM/6000 server on a single node, which acts as /usr server for other nodes. All /usr clients can then execute the ADSM/6000 server code if the correct setup has been provided. Since the installation of a full AIX system is very straight forward, this section describes how to install the ADSM/6000 server on a /usr server system and make the code available for several /usr clients.

The first step is to install the ADSM/6000 server code on the /usr server system, which can be the SP2 control workstation of the IBM POWERparallel Systems SP2, but any node in the IBM POWERparallel Systems SP2 could play this role as well. This procedure is explained in 6.1.1, “Installation on a Full-AIX Node” on page 35 and will not be repeated here.

Then you must separate the customization files from the code and place them onto a local file system. As an example, let us assume you have installed the ADSM/6000 server code on the SP2 control workstation, and you intend to run ADSM/6000 servers on node11 and node12. This means, that node11 and node12 can use the executables from the /usr file system on the SP2 control workstation, but they must have access to locally customized files. This is achieved with symbolic links. We copy the files *dsmserv.opt* and *dsmserv.dsk* to the local file system /var/adsmserv and provide symbolic links to these files. This way, node11 and node12 will each use their own customization files.

These are the steps that have to be performed after the ADSM/6000 server code has been installed on the SP2 control workstation to run the ADSM/6000 server on node11.

1. Run the *dsmserv install* command on the SP2 control workstation to create a dsmserv.dsk in the /usr/lpp/adsmserv/bin directory file.[3]

```
cd /usr/lpp/adsmserv/bin
dsmserv install 1 /var/adsmserv/log /var/adsmserv/db
```

2. Move the files dsmserv.opt and dsmserv.dsk to the /var/adsmserv directory, and set up symbolic links to these files (see Figure 32 on page 105).

[3] We assume that the database and the log have been formatted already and that the directory /var/adsmserv exists.

```
mv /usr/lpp/adsmserv/bin/dsmserv.dsk /var/adsmserv/dsmserv.dsk
mv /usr/lpp/adsmserv/bin/dsmserv.opt /var/adsmserv/dsmserv.opt
ln -s /var/adsmserv/dsmserv.opt /usr/lpp/adsmserv/bin/dsmserv.opt
ln -s /var/adsmserv/dsmserv.dsk /usr/lpp/adsmserv/bin/dsmserv.dsk
```

3. Log on to node11, which uses the same /usr file system as the SP2 control workstation but has its own local /var file system.

4. Format the database and log on the local file system and initialize ADSM/6000.

```
cd /usr/lpp/adsmserv/bin
dsmfmt -log -m /var/adsmserv/log 20
dsmfmt -db -m /var/adsmserv/db 20
dsmserv install 1 /var/adsmserv/log 1 /var/adsmserv/db
```

   Since the symbolic links are already established on the /usr file system of the SP2 control workstation, the initialization will create a dsmserv.dsk file on /var/adsmserv.

5. The ADSM/6000 server is ready to start on node11.

```
cd /usr/lpp/adsmserv/bin
dsmserv
```

6. To register the ADSM/6000 licenses for each of the nodes where you want the ADSM/6000 server running, you must request a license for each of these nodes. The license is dependent on the CPU number of the target machine. When you want to register the license, you have to edit the file /usr/lib/netls/conf/nodelock on the /usr server system and add one line for each node where the ADSM/6000 server should run. The content of the file looks something like:

```

63d9e18fb3cb.02.09.73.01.02.00.00.00 g4vjjf5839d2hwbpvtje3cvphwja ″25GB,25CL″  ″1.2″

63d9e18fb3cb.02.09.73.01.02.00.00.00 7jye4p8mbkwhakgqiwsciiv9c2ja ″25GB,25CL″  ″1.2″

```

   Do not try to use the *register license* command from ADSM/6000 server systems that are clients with respect to the /usr file system. This command fails because you usually have no write access to the /usr file system from the SP2 control workstation and that is where the *nodelock* file resides.

The same procedure could be set up on any node of the IBM POWERparallel Systems SP2 that should run as an ADSM/6000 server and that uses the /usr file system from the SP2 control workstation. Figure 32 on page 105 shows how the symbolic links are set from the /usr server to the /usr client to be able to use local ADSM/6000 server customization files.

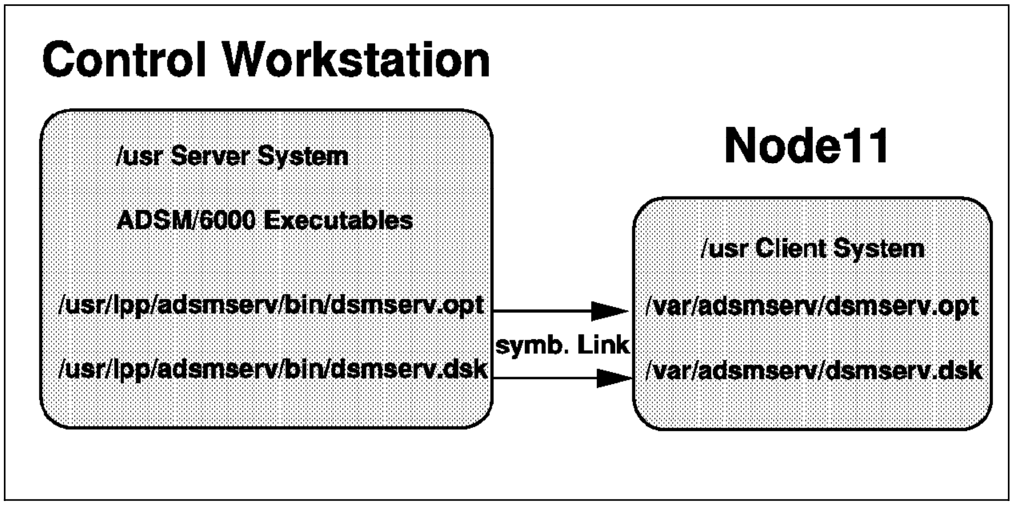

Control Workstation
/usr Server System
ADSM/6000 Executables
/usr/lpp/adsmserv/bin/dsmserv.opt
/usr/lpp/adsmserv/bin/dsmserv.dsk
symb. Link
Node11
/usr Client System
/var/adsmserv/dsmserv.opt
/var/adsmserv/dsmserv.dsk

Figure 32. Use of Symbolic Links for Node Customized Files

12.2 Using Multiple ADSM/6000 Servers

Let us look at the ADSM/6000 client side and see how the client must be set up to address more than one ADSM/6000 server. The *dsm.sys* file of the client needs to have several entries that define the parameters for different ADSM/6000 servers. See the following example of *dsm.sys*, which has stanzas for three ADSM servers.

```
SErvername  sp2sw02
   COMMmethod         TCPip
   TCPPort            1500
   TCPServeraddress   sp2sw02.itsc.pok.ibm.com
   TCPB               32
   TCPWindowsize      640 (*)
   passwordaccess     generate
   inclexcl           /var/adsm/inclexcl.list
   schedlogname       /var/adsm/dsmsched.log

SErvername  sp2sw11
   COMMmethod         TCPip
   TCPPort            1500
   TCPServeraddress   sp2sw11.itsc.pok.ibm.com
   TCPB               32
   TCPWindowsize      640 (*)
   passwordaccess     generate
   inclexcl           /var/adsm/inclexcl.list
   schedlogname       /var/adsm/dsmsched.log

SErvername  sp2cw0
   COMMmethod         TCPip
   TCPPort            1500
   TCPServeraddress   sp2cw0.itsc.pok.ibm.com
   TCPB               32
   TCPWindowsize      24
   passwordaccess     generate
   inclexcl           /var/adsm/inclexcl.list
   schedlogname       /var/adsm/dsmsched.log
```

(*) The 24KB upper limit for TCPWindowsize is cancelled by PTF 20258. See 13.1, “Tuning Performance for the HPS” on page 111 for more information about the TCPWindowsize allowed with this PTF.

This example shows three entries for ADSM/6000 server systems. These are node02, the SP2 control workstation and node11. Several methods can be used to determine which server to contact. One way is to edit the *dsm.opt* file and enter a line such as “servername sp2sw11.” Since ADSM/6000 allows you to set up a *dsm.opt* file for each user, controlled by the DSM_CONFIG environment variable, each user could therefore automatically reach his assigned ADSM/6000 server system. The other method is to use an option on the command line, such as in the following example:

```
# dsm -servername=sp2cw0
# dsmc -servername=sp2sw11
```

Users on the ADSM/6000 client node can concurrently invoke sessions with all ADSM/6000 servers. There is no need to change the port address if several sessions run at the same time. It is also interesting to note that all three sessions can run with the *passwordaccess generate* parameter, which we find easier to use. Have a look in the /etc/security/adsm directory, and you will see that there are three files, one for each session.

12.3 Planning for Multiple ADSM/6000 Servers

The use of multiple ADSM/6000 servers raises several questions that are different depending on how you use the IBM POWERparallel Systems SP2. We make some suggestions, but we are definitely not addressing all the issues related to such an environment.

When planning how to split several ADSM/6000 servers for their optimal use, the first idea that comes to someone′s mind is to simply define that node00 to node10 use the first ADSM/6000 server, node11 to node20 use the second, and so on. That does not work because users and processes are scheduled to different nodes automatically, and they don′t know about it. A user, who is scheduled to node02 at one time would not know where to search for the data when he is scheduled to node12 at another time.

The first proposal binds NFS dataservers to ADSM/6000 servers, such as outlined in Figure 33 on page 107.

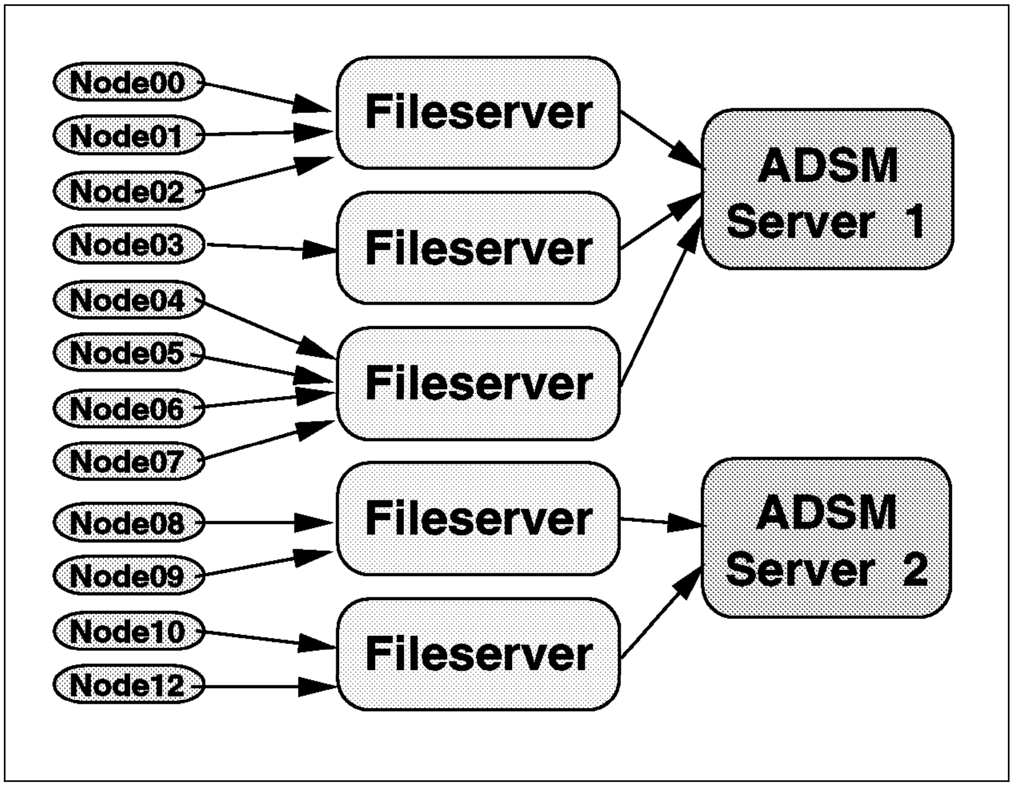


Figure 33. Multiple ADSM Server with Dataserver Oriented Binding

This solution assumes that all backup/restore and archive/retrieve operations are performed from the dataservers. Automatic schedules set up from ADSM/6000 should be prepared to run on the dataservers. If the user needs to perform an ADSM/6000 client operation, he must first log on to the dataserver. An advantage of this solution is that the data traffic is reduced because data travels from the ADSM/6000 client directly to the ADSM/6000 server not involving any NFS traffic. However, users must know about dataserver and must have login capability to the dataservers.

Another approach is to use the virtual node concept. Each user is assigned to a virtual node, and when he opens a session with the ADSM/6000 server, he provides the *-nodename=virtual_node_id* parameter. This would actually allow splitting the use of several ADSM/6000 servers between groups of virtual nodes (Figure 34 on page 108), or we can say groups of users.

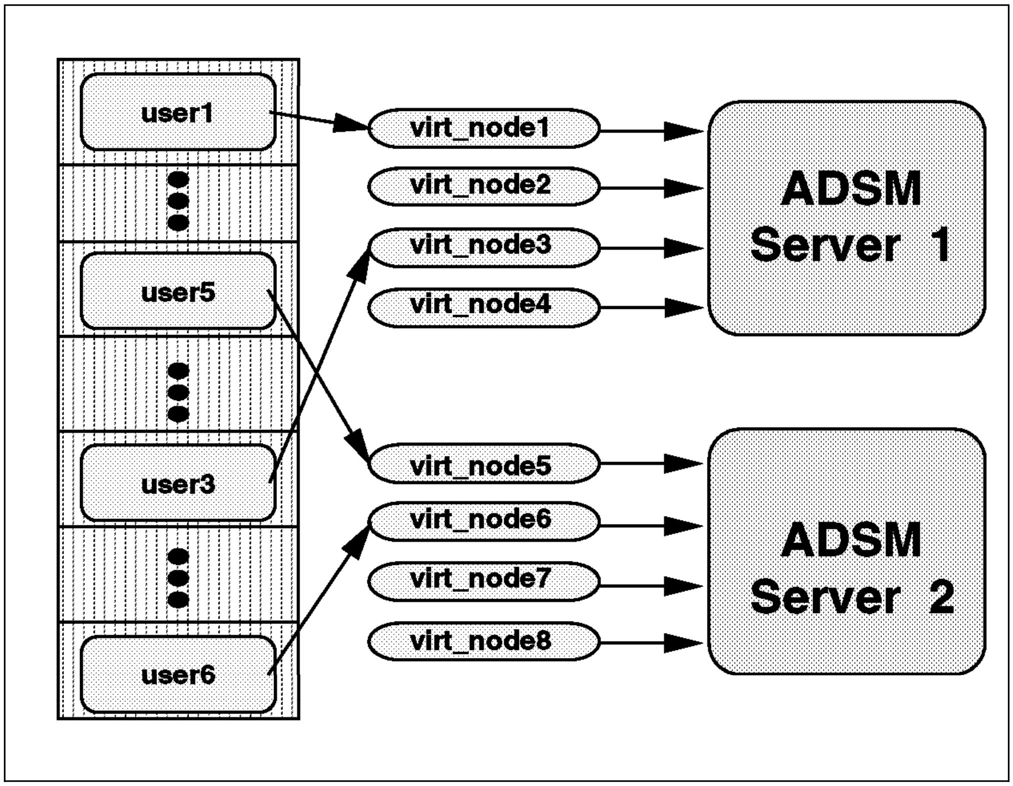


Figure 34. Multiple ADSM Server with Virtual Node Oriented Binding

This solution requires that all users have their personal virtual node ID along with a password. The ADSM/6000 administrator may have to register many virtual nodes in such a case, but the data is uniquely identified.

The last approach uses virtual nodes as well, but this time we bind the application data to a virtual node, and then split the use of virtual nodes between several ADSM/6000 servers. Data from application 1 would be backed up to ADSM/6000 server 1, and data from application 2 and 3 would be backed up to ADSM/6000 server 2. This would work from every node, but it requires careful planning. See also Figure 35 on page 109.

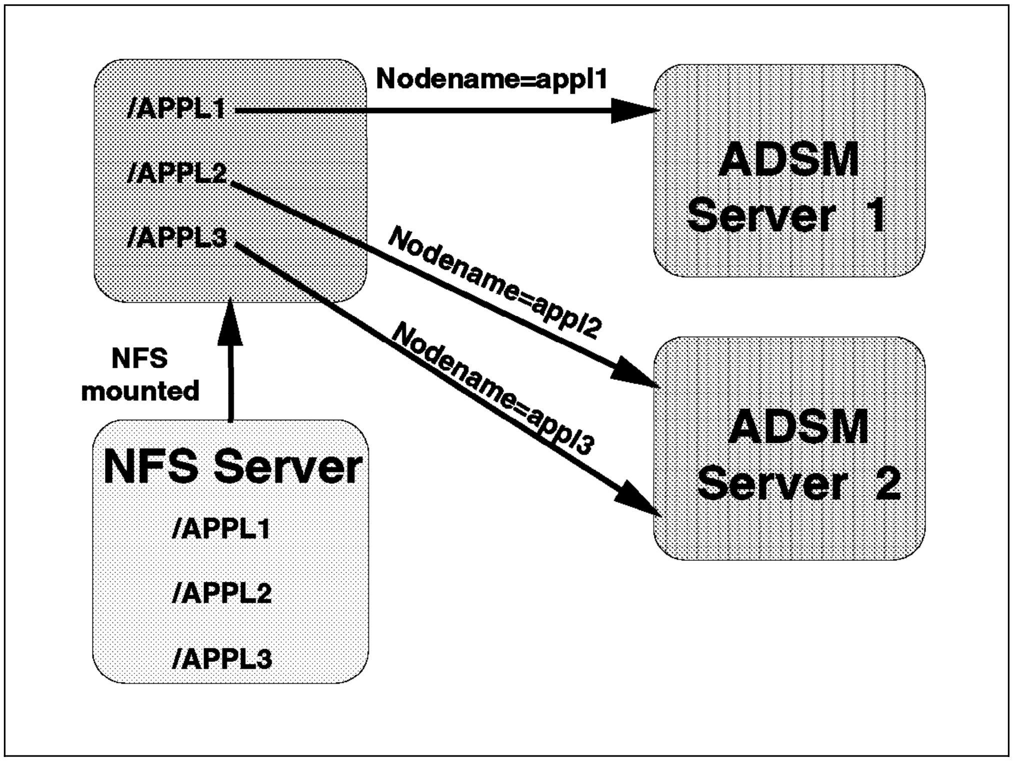


Figure 35. Multiple ADSM Server with Application Oriented Binding

This solution has a clear assignment of data from a specific application to one and only one ADSM/6000 server. It would allow you to plan for different ADSM/6000 servers that support devices according to the needs of a specific application (for example, optical devices for image applications). The data of an application is always stored with an identity of "nodename=application," and a privileged user could retrieve this data from any node.

Section 13. Performance

This section describes the issues we have encountered related to ADSM performance on SP2. Two areas are addressed:

- High-performance switch tuning
- TCP/IP tuning

The following topics address both issues. We expect most SP2 administrators and users will perform their backup and archive operation through the high-performance switch (HPS) to take advantage of its speed, so the issues are intimately related.

13.1 Tuning Performance for the HPS

The communication performance as seen from the application point of view depends heavily on the tuning of the involved communication protocols as well as on the way the application itself communicates. For a given application, the system administrator can tune the parameters for the protocols involved in the communication to achieve the maximum performance. However, one must recall that an optimal tuning for a given application can be sub-optimal for another, even if the two applications perform similar functions. In fact, that can happen with ADSM and FTP, as we will see.

The following is a list of the software involved in the communication between the ADSM client and the ADSM server over the high-performance switch:

- ADSM communication
- TCP/IP sockets
- IP interface to css0
- HPS device driver
- CSS protocol

Each of the following subtopics deals with one of the communication protocols involved, except for the CSS protocol that provides no way to tune its performance.

13.1.1 ADSM Communication

ADSM has its own communication system on top of the TCP/IP socket interface. ADSM provides two tunable parameters:

- TCPBuffer
- TCPWindowsize

The current ADSM/6000 release provides maximum values that are respectively 32KB and 24KB. PTF 20258 cancels the 24KB upper limit for `TCPWindowsize` and allows you to select a 64KB mtu on the high-performance switch. When this PTF is applied, the optimal values of those parameters for ADSM/6000 Version 1 Release 2 on the HPS are respectively 32KB and 640KB. So, you can choose the following parameter values:

- `TPCNODELAY    = yes`
- `TCPBuffer     = 32`

- TCPWindowsize = 640

13.1.2 TCP/IP Communication

The TCP/IP protocol suite has many tunable parameters. They are displayed along with their current value by the *no -a* AIX command. Following is a list of the parameters whose value can affect the ADSM performance over the HPS:

Table 3. TCP/IP Parameters

Option	**Meaning**
tcp_sendspace	default socket buffer sizes for sending TCP data
tcp_recvspace	default socket buffer sizes for receiving TCP data

Those parameters should get values quite different from their default values to optimize ADSM performance on the HPS. The following table shows the default, and recommended value for each parameter:

Table 4. TCP/IP Parameters Setting for the HPS

Option	**Default Value**	**Recommended Value**
tcp_sendspace	16384	655360
tcp_recvspace	16384	655360

Please note that there are equivalent parameters for UDP communication, but changing them will not affect ADSM as it does not use UDP sockets for file transfer. Recall also that the recommended values are expected to optimize the ADSM performance for large file transfers. All values are the highest allowed values for AIX TCP/IP.

13.1.3 IP Interface to CSS

The interface between IP and CSS is configured with the standard AIX command *ifconfig*. The interface parameter that can most affect performance is the *mtu* (maximum transfer unit) size. The mtu for the css interface is displayed by the *netstat -I css0* command. For maximum ADSM performance, you should apply the PTF 20258 and set it to 64KB with the command *ifconfig mtu 65536*.

13.1.4 HPS Device Driver

The performance of the device driver code is affected by the availability of some system resources. The most relevant resource is the buffer space available. That is controlled by some parameters that can be displayed with the *no -a* command. The following table shows the most relevant ones:

Option	**Meaning**
lowclust	low water mark for the cluster mbuf pool
lowmbuf	low water mark for the mbuf pool
thewall	maximum amount of memory that is allocated to the mbuf and cluster mbuf pools
mb_cl_hiwat	high water mark for the cluster mbuf pool

Those parameters should get values quite different from their default values to optimize ADSM performance on the HPS. The following table shows the default value and recommended value for each parameter:

Table 5. System Buffers Parameters Setting

Option	**Default Value**	**Recommended Value**
lowclust	45	100
lowmbuf	104	272
thewall	2048	16394
mb_cl_hiwat	90	1220
sb_max	65536	1310720

Recall that the recommended values are expected to optimize the ADSM/6000 performance for large file transfers.

Chapter 5. User Access Issues

In this chapter we describe some issues related to the use of ADSM/6000 in the IBM POWERparallel Systems SP2. The way the IBM POWERparallel Systems SP2 works, users and processes are scheduled to different nodes on the IBM POWERparallel Systems SP2 without their explicit knowledge on which node they are running. The use of NFS with *amd* they are scheduled. These facilities imply particular actions from the ADSM administrator, which are described in Section Section 14, "File Names."

Section 14. File Names

This section proposes and describes the specific solutions to particularities of implementing ADSM on the SP2 environment. When using these recommendations, the administrator will avoid inconsistent file names given by ADSM to backup and archive files, and maintain the function provided by amd that allows users to transparently work on any SP2 node.

Information about these problems and suggested solutions is given in the following topic.

- Symbolic links
- NFS access

14.1 Symbolic Links

ADSM stores a file using the file′s absolute pathname. To do that, if a user references a file with a relative pathname, ADSM has to prefix the working directory pathname to the file relative pathname. This procedure is necessary to let ADSM be independent of the working directory of the user.

Let′s see what happens if a user wants to save a file named *filename* located in his current working directory. Let′s suppose that the user is working in his home directory. In the AIX environment, a user′s home directory usually is a symbolic link to /home/*username*. If the user references the file with its relative pathname (*filename* in our case), the file is stored with its absolute path name after the resolution of all the symbolic links in the working directory path name. Unfortunately, the working directory is a symbolic link, and it will be replaced by another name - the name found by resolving the symbolic link. In our case, the name is /home/*username* instead of /u/*username*, as the user expects. Then the file is stored in ADSM as /home/username/*filename*.

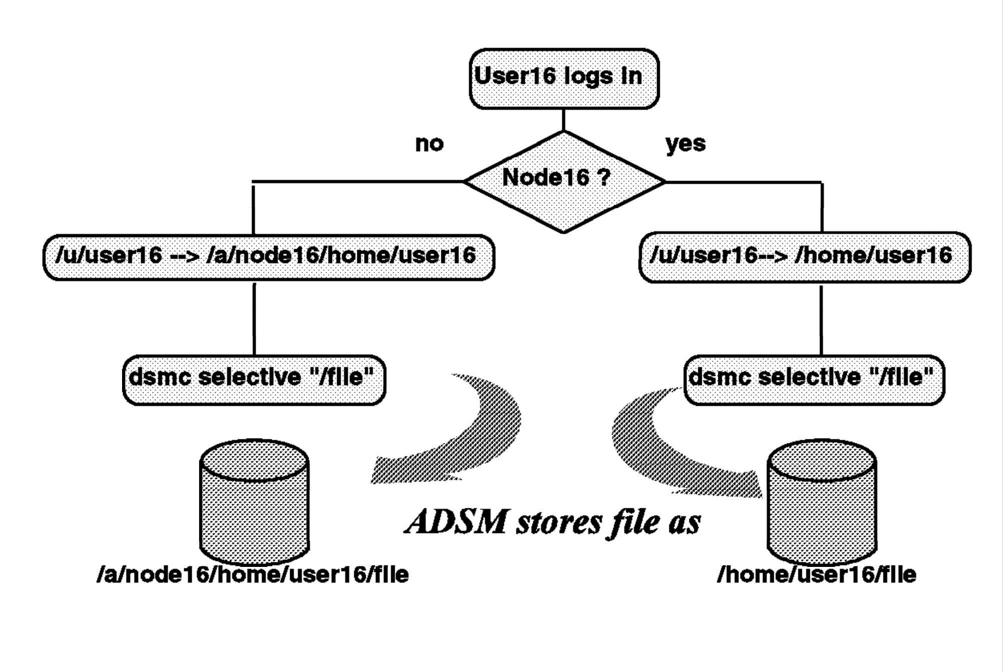


Figure 36. Example of Symbolic Link Handling in ADSM

In the SP2 environment, if *amd* is used, the story is more complex: the files in /u are symbolic links, dynamically created by *amd*. Usually they point to /home/*username* for users defined to have a local home directory. Instead, they point to /a/*NFSservername*/home/*username* if the user′s home directory is remotely mounted through NFS from *NFSservername* where it resides in /home. For example, on ITSO Open Systems Lab.′s SP2, user *user16* has his home directory physically located on *node16*, which is defined to be the file server for him. When user *user16* logs onto node *node12*, he sees his home directory located in /a/node16/home/user16, as it is automatically mounted through NFS by *amd*. Of course, *user16* logged onto *node16* has his home directory located in /home/user16. So, thanks to *amd*, he does not care (and usually does not even know) where the files are. In fact, on both the nodes (and effectively on all the nodes of SP2) he can always access his file as /u/user16/*filename*. In our example, the user file will be stored as /a/*NFSservername/home/username/filename* provided the user is referencing the file with its relative pathname.

14.1.1 User′s Point of View

From the user′s point of view, the issues discussed here lead to the following consequences:

- The user cannot store and access his data by using the relative pathname he is accustomed to (in our example, it is the relative path filename *userfile*).
- The user must use the name provided by ADSM, which can look quite strange to him, to be able to retrieve the file.
- The name is different if the user accesses the file using a relative or an absolute filename.

14.1.2 ADSM Usage to Bypass the Symbolic Links

The easiest way to bypass the problem of the symbolic links is to always reference files with their absolute pathname. That procedure allows the user to use any symbolic link and reference his files with a name he is accustomed to using. Of course, it is not very convenient, especially for long pathnames.

14.2 NFS Access

ADSM treats the files from NFS and JFS mounted filesystems as equivalent. That approach does not help to manage the NFS configuration, where a single NFS server can serve many clients. In fact, it causes two problems:

- File identity problem: the same file appears to ADSM as many different files.
- Performance problem: the file travels twice over the network.

We address those issues in the following topics.

14.2.1 NFS Files Identity

One file residing on a file system exported by the NFS server and saved to ADSM by different clients will appear to ADSM as different files. For example, the file stored on the NFS server in /home/*username*/*filename* when archived from node *node16* will look to ADSM like node16:/a/*NFSservername*/home/*username/filename* (recall our previous discussion of *amd* symbolic links). The same file stored from node *node15* will be treated by ADSM as node15:/a/*NFSservername*/home/*username/filename*; that is different from the other name. Usually, the same one file will appear to ADSM as many different files - as many as the clients. This behavior is a problem on SP2 where a large part of the file system of each node is usually accessed through NFS from a server (either another node or an external machine).

14.2.2 NFS Performance

Backing up NFS mounted files can have a relevant negative impact on the performance of:

- ADSM backup operation: NFS speed is significantly lower than the disk speed or the tape speed, and also lower than ADSM speed through the high-performance switch.
- Other network operations: the network is also used for other NFS traffic along with remote access, and so forth.

Therefore, we do not recommend performing the backup operation from the NFS client. The most efficient solution surely is to perform backups directly from the NFS server to the ADSM server. But ADSM provides no easy way to reschedule the execution of the backup operation from the NFS client to the NFS server. For all those reasons, the system administrator will schedule backups to be executed from the NFS server. But the users will typically run their archive or backup commands from a node.

14.2.3 ADSM Configurations to Solve NFS Problems

We can have different approaches to the problem of NFS file names. In the present topics, we outline some viable solutions to the NFS files naming problem. All the solutions involve only the ADSM configuration. Another approach, that involves some shell programming, is discussed in the following topics.

1. Use a *virtual node*. ADSM accepts the definition of nodes that do not physically exist. Each user can be assigned to a virtual node along with a password. Whenever the user wants to work with ADSM/6000, he uses this virtual node identification when he starts a session with the ADSM/6000 server. This can easily be achieved by adding an option of *-nodename=virtual_node_name* to the *dsm* or *dsmc* command, such as in the following command:

   ```
   # dsmc -nodename=hans
   ```

   By using the virtual node with ADSM/6000, a user can be logged in to any node of the SP2, but to the ADSM server he will always look like the virtual node name.

   Advantages of this method are:

   - NFS files are uniquely identified across all the client nodes.

   The disadvantages are:

   - The ADSM manager has to define to ADSM a virtual node for each SP2 user, and there can be many.
   - The ADSM operation affects the network traffic more than necessary, because files travel from the SP2 file server to the node where the user is logged in and then to the ADSM server. The performance is lower.
   - It is very likely that there are automatic incremental backup procedures running from the SP2 file server, and the NFS files have different identities when saved from the client nodes instead of the file server.

2. Another approach is to have the users perform all their ADSM operations from only one node. This requires that users force a logon to a specific node before using ADSM/6000 client services.

   Advantages of this method are:

   - NFS files are uniquely identified across all the client nodes.

   The disadvantages are:

   - It is very likely that there are automatic incremental backup procedures running from the SP2 file server, and the NFS files have different identities when saved from the client nodes instead of the file server.
   - The ADSM operation affects the network traffic more than necessary, because files travel from the SP2 file server to the node where the user is logged in and then to the ADSM server. The performance is lower.
   - The user must explicitly login to the selected node to perform any ADSM operation.
   - The users have access to other ADSM data saved by that node.

3. Similar to the second approach is to have the users perform all their ADSM operations from the NFS servers. This requires that the users actually login to the SP2 NFS file server and perform the ADSM operations on that node.

   Advantages of this method are:

   - The files are uniquely identified.
   - Network traffic is not impacted more than necessary because the files only travel from the NFS file server to the ADSM server.

   Disadvantages are:

   - The users must have access to all NFS servers, and they must know which is the NFS server for each file.

4. And the final suggestion is to give the users the ADSM password for the NFS servers, but have them execute the ADSM operation from the node where they are logged in currently.

   Advantages of this method are:

   - The files are uniquely identified.

   Disadvantages are:

   - The users have access to data that belongs to other users because all users identify themselves to the ADSM/6000 with the same node ID and password. The users must also know which is the NFS server for each file.
   - The ADSM operation affects the network traffic more than necessary, because files travel from the SP2 file server to the node where the user is logged in and then to the ADSM server. The performance is lower.

You can see there is no unique solution to the naming problem. The best solution for a given environment depends essentially on:

- Performance requirements
- Security requirements
- Complexity of the system

14.2.4 Example of Use of Virtual Nodes

Let us look at an example of how the concept of virtual nodes is used in the SP2 with ADSM/6000. We assume that the SP2 control workstation is the NFS file server for all /home directories. The SP2 control workstation runs automatic periodic backups of all files in the /home directory driven by the ADSM scheduler. An ADSM/6000 administrator should supervise the successful completion and correct any anomalies.

Each end user is assigned to one virtual ADSM/6000 node and receives a password along with the virtual node. He uses this virtual node to identify himself when he starts a session with the ADSM/6000 server. The use of virtual nodes in ADSM/6000 helps to establish a binding between files in the ADSM storage base and end users. From the view of the ADSM/6000 server, virtual nodes are no different from physical nodes.

The end users are logged into different nodes of the SP2 when they log in without knowing on which node they are actually working. They use their /home directories through NFS which mounts automatically through the *amd* automounter. To allow the end users to restore files that have been stored by the automatic backup procedure on the SP2 control workstation, the ADSM/6000 administrator must grant access rights to nodes and users who want to perform restore operations. This is performed with the following commands on the SP2 control workstation.

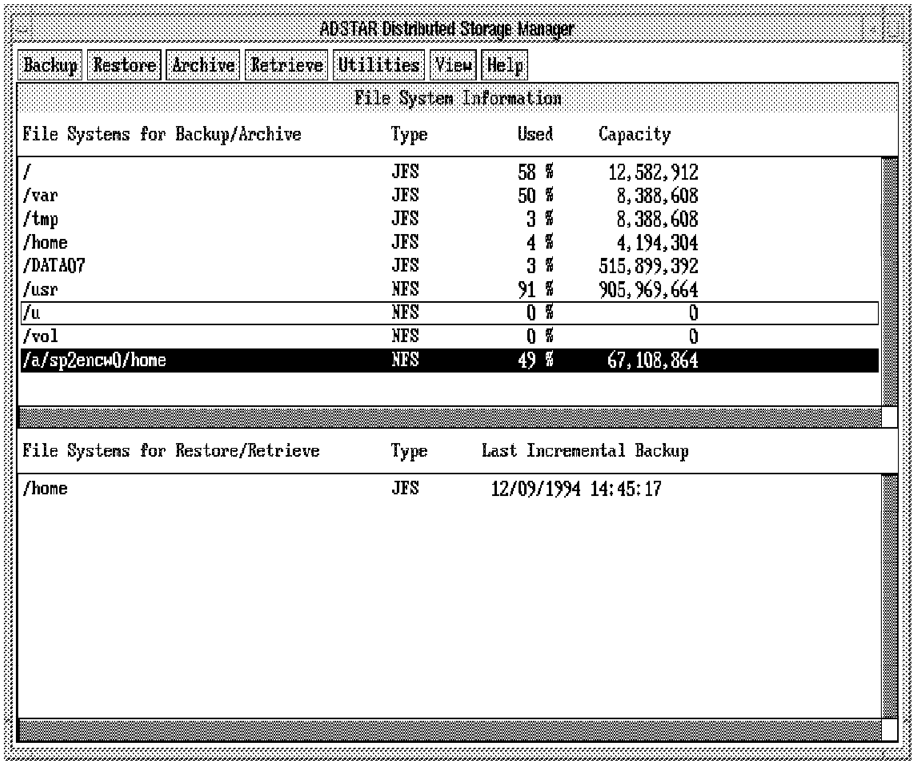
```
# dsmc set access backup ″/home/user1/*″ virt-user1 ″*″
# dsmc set access backup ″/home/user2/*″ virt-user2 ″*″
   ⋮
# dsmc set access backup ″/home/usern/*″ virt-usern ″*″
```

The string “virt-user1” is the virtual node ID that you have assigned to user1, and “virt-user2” is the virtual node ID assigned to user2, and so forth.

When the end users who are logged into nodes of the SP2 wish to access the data that has been backed up from the SP2 control workstation, they must use their virtual node ID when starting a session with ADSM/6000.

```
# dsm -nodename=virt-user-ID
```

After starting the session with the ADSM/6000 server, select the “Set user/node name for Restore/Retrieve” button, which can be found in the utility menu. Then change the node name to the name of the SP2 control workstation, which is the NFS server system in our case, and set the user name to whatever your current user ID is.



File Systems for Backup/Archive	Type	Used	Capacity
/	JFS	58 %	12,582,912
/var	JFS	50 %	8,388,608
/tmp	JFS	3 %	8,388,608
/home	JFS	4 %	4,194,304
/DATA07	JFS	3 %	515,899,392
/usr	NFS	91 %	905,969,664
/u	NFS	0 %	0
/vol	NFS	0 %	0
/a/sp2encw0/home	NFS	49 %	67,108,864

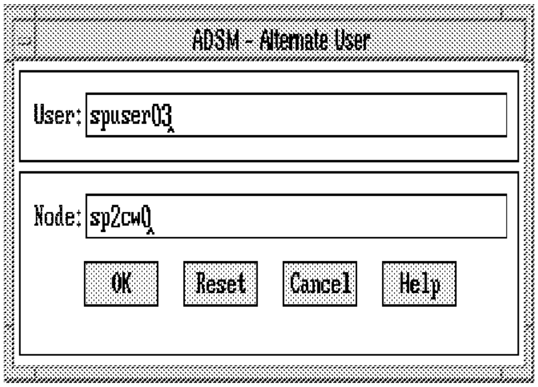
File Systems for Restore/Retrieve	Type	Last Incremental Backup
/home	JFS	12/09/1994 14:45:17

Figure 37. Backup/Restore GUI with /a/sp2encw0/home File System

You should now have access to the file system backed up from the SP2 control workstation and to the data in that file system where you are the owner. If you actually restore files, it is necessary to change the destination of the files because otherwise, ADSM will try to create the directory structure as it was seen from the SP2 control workstation. It is usually sufficient to enter a destination directory of ./newdir where ADSM can place the restored files. You, as the owner of these files, can then move them to the desired location.

If you want to perform any backup or archive commands yourself from one of the SP2 nodes, first reset the user identification so that ADSM sees you as node *virt-user*. Use the **Reset** button on the screen shown in Figure 38 for this purpose. If you want to back up or archive files from your /home directory, select the file system /a/sp2encw0/home. This is the file system where the automounter mounts the /home directory from the NFS server.

Unfortunately the operation described in this section results in two sets of files for a particular user in the ADSM/6000 database. One is the set of files that ADSM/6000 keeps with an attribute of belonging to the SP2 control workstation. The other set are those files that ADSM/6000 keeps as belonging to the virtual node ID of users.


ADSM - Alternate User

User: spuser03

Node: sp2cw0

Figure 38. Change User Identification GUI

14.2.5 ADSM Client Interface

Another (partial) solution to the issues just discussed is to write an interface that performs the following functions:

1. Replace each file reference in the user command with an absolute filename that contains no symbolic link.
2. Reschedule the execution of any user command to the NFS server for the referenced files.

Those functions can be achieved with a few simple shell scripts. We have written such scripts, and they are listed in Appendix B, “The adsmc Interface” on page 147. The advantages of this solution are:

- The reference to a file is independent of symbolic links.
- The reference to a file is independent of the node.

While the disadvantages of this solution are:

- The user must have remote login capability to the NFS server.[4]
- There is some overhead in processing complex directory trees.
- The semantics of file access are slightly different.

[4] The *Sysctl* function of IBM AIX PSSP Version 1 Release 2 can solve this.

Chapter 6. ADSM Availability

This chapter focuses on the availability of the ADSM/6000 server and the data which the ADSM/6000 server works with. We discuss several methods to increase the availability of the ADSM/6000 server. Among them are:

- Mirroring the ADSM/6000 database
- Dumping the database at regular intervals
- Auditing the storage pools
- Exporting the ADSM/6000 data and database
- Running a second ADSM/6000 server
- Salvaging the database

Section 15. ADSM Availability Issues

ADSM/6000 is a service provided from a central point to backup and archive data from distributed systems and to lower the risk of losing data on individual workstations. The question is "How much should be invested to backup the backup system itself?" For its operation, the ADSM/6000 server uses data and code that may get lost through physical damage or logical corruption. Most notable are the database and the log file used by the ADSM/6000 server, which are highly critical for the operation. The database holds information such as:

- Information about files being backed up or archived
- Information about registered nodes
- Information about management policies
- Information about schedules
- Information about event records
- Information about activity logs
- Information about ADSM volumes

Losing the database means losing any way of accessing the data in the storage pools. There is no way of rebuilding the database from the information found in the storage pools. So read the following note carefully.

Note: Go through any affordable effort to keep the database consistent and operational!

The database must, therefore, be protected against logical inconsistencies and physical damage. There are other components besides the database that the ADSM/6000 needs for its operation. Let us look at a summary of all components that are involved in the operation:

- The database
- The recovery log
- The configuration files
- The ADSM code itself
- The data from the storage pools

The recovery log ensures that the database remains consistent by recording all changes made to the database. Any update to the database can be regarded as a transaction, and the information concerning this transaction is logged in the recovery log. If the database operation is interrupted, the transaction remains as an uncommitted item in the recovery log, and this information is used to rollback the transaction when the database becomes online again. The recovery log is vital to the operation of the ADSM/6000 server almost as much as the database itself. If you lose the recovery log, the ADSM/6000 server will not start, even if the database is clean.

In fact, all of these components mentioned in the list above should be protected, but this is a trade-off between availability and cost.

15.1 Mirror the Database and Log

One way of securing the database against physical loss is to mirror the database. Mirroring is the process of writing the same data to multiple storage devices at the same time. Mirroring does not provide protection against logical corruption of the database, since logical errors will be propagated to all mirrors as well.

With ADSM/6000 on the SP2, there are two methods of mirroring the database. One is by taking advantage of the logical volume mirroring feature of the AIX/6000 operating system; the other is by using the ADSM/6000 server built-in mirroring feature. The recommendation is to use the ADSM/6000 mirroring feature, since it offers better synchronization functions adjusted to ADSM/6000.

When you use the ADSM/6000 mirroring function, the server automatically places the failing volume offline if a media failure occurs, and continues with the remaining volumes. Once the failed disk is repaired and operational, the volume must be made available to the ADSM/6000 server, which automatically synchronizes the volume again. The process of how to set up the mirrored volumes for the database is explained in *Getting Started with ADSM/6000*.

Some things that you want to look for when creating the mirroring volumes are:

- Place the mirrored volumes on distinctive physical disks. It may not be obvious how to do this, since AIX uses a logical volume architecture, and logical volumes can be spread across several physical volumes. The easiest way to achieve this is to create new logical volumes in AIX, assign distinct physical disks to each AIX logical volume, and define them as ADSM/6000 mirrored database volumes.
- The database volumes should have positions on the disk that guarantee the best access times. These are positions close to the center of the disk. Again, positioning can be specified when creating the AIX logical volumes.

You can also expect a slight performance improvement by using database mirroring for read operations. A read operation will be completed from the volume that has the fastest access. On the other hand, for database write operations, you will have to pay a slight penalty since write operations to the database will not complete until all mirrored volumes have completed the write.

The ADSM/6000 log files that work together with the database can be mirrored as well. The database log files will not grow to the size of the database itself. Therefore, if you decide to mirror the database itself, you may as well spend the additional space to also mirror the log files.

Two options, MIRRORREAD and MIRRORWRITE, in the ADSM/6000 server options file specify the mode used for reading and writing database and log pages.

The options for MIRRORREAD are:

MIRRORREAD normal (default)
Specifies that only one mirrored volume is read to obtain the desired page.

MIRRORREAD verify
Specifies that the server reads all mirrored pages and checks for invalid pages on mirrored volumes. This has an impact on performance.

The options for MIRRORWRITE are:

MIRRORWRITE sequential

Specifies that each mirrored volume is written sequentially: each I/O operation is executed after the previous one is successfully checked. This is the default when writing to the ADSM/6000 database. It has an impact on performance.

MIRROWRITE parallel

Specifies that all mirrors are written in parallel. This parameter is the default for the ADSM/6000 log file.

We recommend that you use the defaults.

15.2 Dump/Load the ADSM Database

Another way of protecting ADSM/6000 against loss of the database is to put a copy of the database on a sequential medium, which can be used to recover the database in case of a catastrophic failure. Basically, ADSM/6000 provides the functions DUMP and LOAD for this operation, which are executable as stand-alone utilities. The dump also has a version that can be executed online from the administrator console.

15.2.1 How to Dump the Database

The preferred method of dumping the database is to use the stand-alone utility while the ADSM/6000 server has been halted because it does not introduce the problems involved in dumping a database that has ongoing transactions. To be able to use the stand-alone dump utility, you must define a “device class” so that the dump medium is determined. Look at the following example, which displays a section from the dsmserv.opt file (usually found at the end of the file).

```
DUMPLOADDB library manlib libtype=manual
DUMPLOADDB devclass tapeclass devtype=cartridge library=manlib
DUMPLOADDB drive manual 3490drive device=/dev/rmt0
```

These three lines in the *dsmserv.opt* file are necessary because the device classes defined with the DEFINE DEVCLASS command during the online operation of the ADSM/6000 server are not available. In our example, these three lines control the output of the stand-alone dump command to be written to the IBM 3490E Model E11. Now you can use the following command to dump the database to a scratch tape.

```
DSMSERV DUMPDB DEVCLASS=tapeclass
```

The tape must be labeled with the dsmlabel utility before it can be used for this operation. Once the dump is completed successfully, remove the tape from the drive and store it in a safe place. It is possible that several scratch tapes are used for the dump, and you should carefully record the order in which the tapes where used during the dump. This is important for the recovery process.

The other way of dumping the database is to do it online. The commands must be executed by the ADSM/6000 system administrator either from the ADSM console or from an administrator client interface (command line or graphical interface). Using the online method does not require a definition of a device class in the dsmserv.opt file, but you will use a device class that is defined for the active and running ADSM/6000 server. Let us look at an example where we use a device class that writes sequential files on the AIX file system. Define the following device class named *dbfiles*.

```
adsm> DEFINE DEVCLASS dbfiles DEVTYPE=FILE DIR=/var/adsm/dumpdb
```

If you execute the DUMP DB command from the active ADSM/6000 server, such as:

```
adsm> DUMP DB DEVCLASS=dumpdb CONSISTENCY=yes
```

it flushes any cached database updates to the database and then dumps the database to the file:

```
/var/adsm/dumpdb/00000001.DMP
```

The problem with the online database backup is that if the ADSM/6000 server is in the process of backing up, archiving, moving files, or any other active operation, the dump of the database will not be consistent, and recoverability is limited. To eliminate this risk, use the parameter CONSISTENCY=yes, as we have done it in the example above. This will stop all transactions for the duration of the database dump and write a consistent database image. Also see the section about auditing a database in 15.3.1, "Audit the ADSM Database" on page 132 to understand how to handle an inconsistent image.

Now let us outline a method that we have implemented on our SP2 in the ITSO Open System Lab. We have set up a procedure that periodically produces dumps of the ADSM/6000 database, preferably to an NFS mounted remote disk. This has the advantage that in case of a catastrophic failure, for example if the disk is lost on which the ADSM/6000 server database is located, we can reload the database from the remote disk. We run the following script daily, controlled through cron, and preferably at a time of low ADSM/6000 server activity.

```
#!/bin/ksh
#******************************************************************
# query the database and log the information to a file which
# can be used by the administrator to check database utilization.
# Before dumping the database, save any existing dumps.
# P. Koepp,  12/01/94
#******************************************************************
logdir=/var/adsm/dumpdb
# query the database and log the info in dsmdb.log
date >> /var/adsm/dsmdb.log
dsmadmc -id=admin -pass=admin q db >> /var/adsm/dsmdb.log
# query the log file and log the info in dsmdb.log
date >> /var/adsm/dsmlog.log
dsmadmc -id=admin -pass=admin q log >> /var/adsm/dsmlog.log
# dump the database
if [[ -s $logdir/00000001.DMP ]]; then
   mv $logdir/00000001.DMP $logdir/00000001.DMP.save
fi
dsmadmc -id=admin -pass=admin dump db devclass=dumpdb consistency=yes
```

Before the dump is written to the 00000001.DMP file, the file must be moved or removed, otherwise the dump will fail. If the dump of the database requires more space than specified by the MAXCAPACITY parameter of the device class, you have to handle additional dump files. Notice also that we write a summary of the usage of the database and the recovery file to a log file that we have set up. These files will help the ADSM/6000 administrator supervise the usage of the database and the log and be alerted if these files have to be extended. It would be nice to parse the output and send a warning message to the administrator if utilization of the log or the database exceeds 80%.

15.2.2 How to Recover the Database

If the ADSM/6000 server volumes are still readable, try to recover the existing database before loading the database from dump volumes. The ADSM/6000 server provides an auditing facility, described in 15.3.1, "Audit the ADSM Database" on page 132, which is also capable of fixing parts of the database. This depends on when your last dump was taken. The newer the dump, the more likely it is to start with loading the recent dump.

Before loading the database from the dump volumes, use the DSMSERV INSTALL command to reinitialize the database and the log files, such as:

```
DSMSERV INSTALL 1 /usr/lpp/adsmserv/data/db 1 /usr/lpp/adsmserv/log
```

If you use files instead of raw logical volumes, it is worth considering formatting these files with the DSMFMT command, even if the files already exist, before using the DSMSERV INSTALL command. Now make sure that the *dsmserv.opt* file contains the correct device class definition to access the dump volumes. For example, if the dump has been written to tape volumes, you need something like the following:

```
DUMPLOADDB library manlib libtype=manual
DUMPLOADDB devclass tapeclass devtype=cartridge library=manlib
DUMPLOADDB drive manual 3490drive device=/dev/rmt0
```

If the database dumps have been written to a device class of FILE, you need a line such as:

```
DUMPLOADDB devclass dumpdb devtype=FILE DIR=/var/adsm/dumpdb
```

Now issue the DSMSERV LOADDB command to reload the database from the dump volumes. For example:

```
DSMSERV LOADDB DEVCLASS=dumpdb
```

The last step to recover the database is the DSMSERV AUDITDB command. This command can be quite lengthy, and it is only required if the database is not consistent. For example, if the database has been dumped without the CONSISTENCY parameter, you must run the DSMSERV AUDITDB command. If you do, then run the DSMSERV AUDITDB command before you start the ADSM/6000 server, such as:

```
DSMSERV AUDITDB FIX=yes
```

After you have reloaded the database, it is important to verify if the information in the database is still consistent with the information in the storage pools. It is recommended to audit the storage pools as described in 15.3.2, “Audit the Storage Volumes” on page 132.

15.3 ADSM Auditing

Since ADSM/6000 is a very complex system running with several subcomponents that must work together, it is important to have some tools to verify the correct teamwork of all subcomponents. There are several auditing functions available in ADSM/6000 used to check for inconsistencies in the total ADSM/6000 operations. These are:

- Audit the database
- Audit the storage volumes
- Audit the library
- Audit the license

The question is: “When should you run these utilities?” They are likely to use system resources, such as CPU seconds, and you want to execute them only when required.

15.3.1 Audit the ADSM Database

Auditing the database is a process that can only be started as a stand-alone utility. There is no equivalent to perform this operation from the ADSM/6000 administrator command line while the database is online. You use this command most likely after you have reloaded a database, as described in 15.2.2, “How to Recover the Database” on page 131. Auditing the database is a must if you have dumped the database with the CONSISTENCY=no parameter, but you should not have to do it when the dump was performed with CONSISTENCY=yes. Since auditing the database can be a very lengthy procedure, we want to point out again how important it is to dump the database in a consistent way.

The command you use to audit the database while the ADSM/6000 server is halted is:

```
DSMSERV AUDITDB Fix=yes
```

15.3.2 Audit the Storage Volumes

If any inconsistencies between the data stored in the storage volumes and the database are likely, run the auditing process on the storage volumes. For example, after reloading the database it is highly recommended to run this procedure. Obviously, you want no activity on the volume while the audit process is running, so make sure that no other background processes, such as “moving data” or “delete volume,” are active and that no ADSM/6000 clients are using these volumes. The process invoked by the audit command can fix any inconsistencies between storage pool data and database entries. This means, that ADSM deletes database records that point to a file that does not exist and deletes any references to a file on a storage pool volume for which there are no database records. Database records are also deleted for files that have I/O errors when accessing the file on the storage pool volume.

Before you invoke this command, look at a summary of all volumes so you know which volumes you want to audit.

```
adsm> q volume
adsm> audit volume /dev/rbackup fix=yes
```

The duration of processing this command depends on the size of the database. Be warned that it can be very time consuming.

15.3.3 Audit the Library

Auditing a library is necessary to check if the information ADSM keeps about a library is consistent with the library itself. For example, if you manually move a tape inside an automated library, this process corrects the server′s inventory of the library. The command is:

```
adsm> audit library 3494lib
```

Run this command only if you suspect inconsistencies or if you actually moved tapes manually inside an automated tape library.

15.3.4 Audit the License

This command gathers information about the license and checks if the license is in compliance with the actual usage of the ADSM/6000 server. For customers, it is useful to run this command after they have acquired and registered a license keyword. Use the command as follows:

```
adsm> audit license
```

15.4 Export/Import Functions

The export/import function of the ADSM/6000 server allows you to copy all or part of the server to a sequential medium. There are two reasons why this can be very helpful in case of a catastrophic failure.

- The exported data can be carried to a remote ADSM server and be imported on that system.
- The exported data can be reimported to the same ADSM server after repairing the failing component.

Comparing the functions of the export/import utility to the database dump/load facility we see that this gives us not only the database contents, but also all of the data in the storage pools. It is most desirable to export the complete server as often as possible to have a full backup of all components of the ADSM/6000 server. However, this process is too costly in terms of time, CPU usage, and storage resources to be applied regularly.[5] It is, therefore, recommended to export only parts of the server, for example, those parts that have a high

[5] The export function in ADSM/6000 is currently redesigned to improve performance.

requirement for not being lost and use other methods in the day-to-day operations to protect the ADSM server from loss of data.

Let us look at two examples of how we used the export function for disaster recovery.

15.4.1 Example One for Export/Import

Let us assume that once a week a company needs to carry off-site the data that belongs to some workstations. This data will flow into the regular backup scheduled for each individual workstation and be resident somewhere in the storage pools of the ADSM/6000 server. How do we collect the data and store it on tape or optical disk so we can carry it off-site?

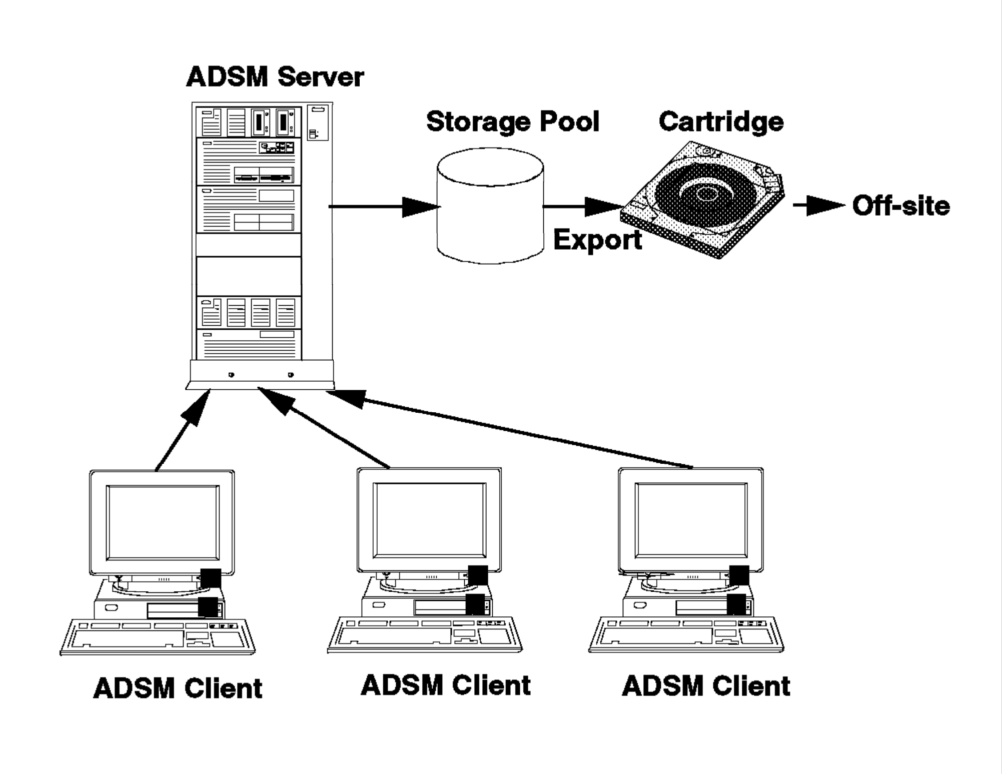


Figure 39. Export/Import Example One

The export command has several options to select only parts of the server data. The options we use for this example are the *nodename* option and the *filespace* option. The *nodename* option allows us to select only a subset of all nodes registered at this server, and the *filespace* option further limits the exported data to only those file systems where the data that we are interested in resides. In addition to the previous options, we will use the *filedata=active* option to export only the active versions of files. This selection of options cuts the overall processing time of the export command to a reasonable amount. The command will look like:

```
adsm> export node node01,node02,node03
                  filespace=/appl14
                  devclass=cartridge
                  scratch=yes
```

This command exports all archive and active backup versions from the file system /appl14 from nodes node01, node02 and node03. The activity log can be used by the administrator to identify the tapes that have been used for the export function, and he may take these tapes and send them off-site.

15.4.2 Example Two for Export/Import

Let us assume in this example that there is a need to recover data from a single node as quickly as possible. The data from this workstation is regularly backed up to ADSM/6000 server 1, but if ADSM/6000 server 1 has an outage, we want to switch quickly to ADSM/6000 server 2 and continue to work with this one until ADSM/6000 server 1 comes back online.

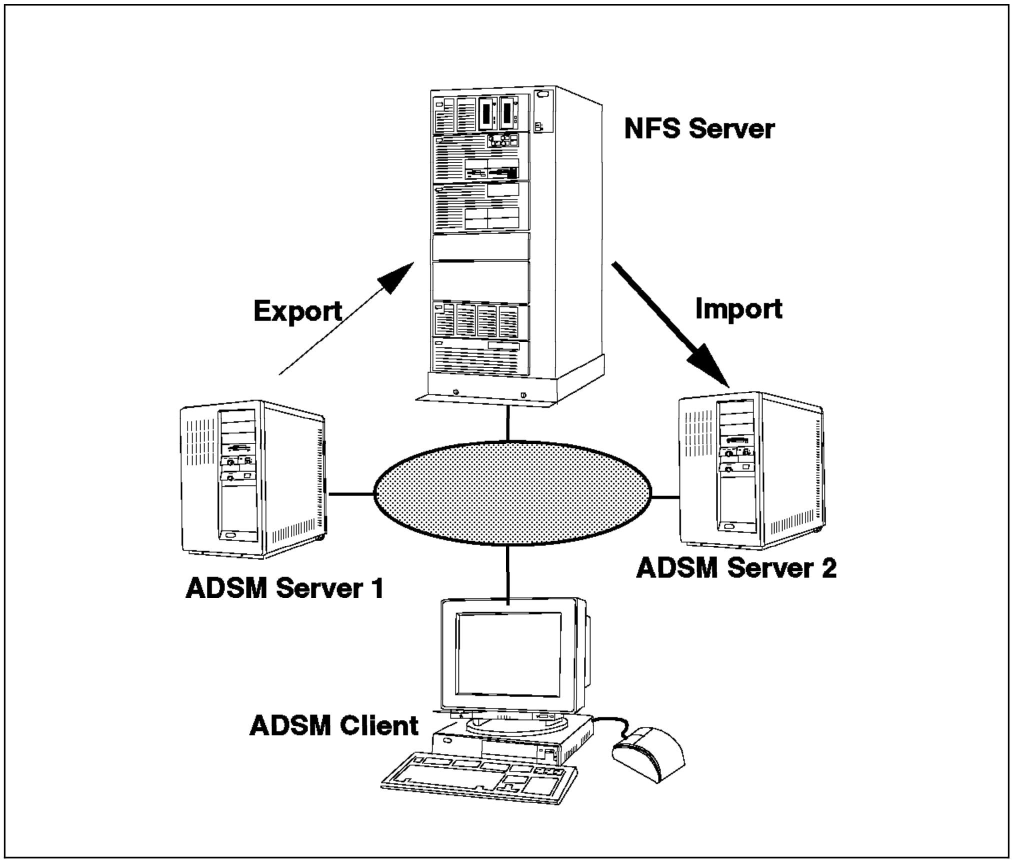


Figure 40. Export/Import Example Two

The export command in this example is used to write all workstation related data to an NFS server which is accessible from ADSM/6000 server 1 and ADSM/6000 server 2. ADSM/6000 server 2 must import the workstation related data into its own database and storage pools.

To be able to export the data to a file system instead of a medium, such as a tape, requires you to define a device class with a type of *FILE*. You can do this by issuing the following command on ADSM/6000 server 1 and on ADSM/6000 server 2.

```
adsm> define devclass FS devtype=file dir=/adsmserv/data
```

The directory /adsmserv/data is located on the NFS server and should be accessible from both ADSM servers. Now you can export the workstation data from the ADSM server 1 with the following command:

```
adsm> export node nodename filedata=all devclass=FS
```

This could be done regularly and be controlled through a cron job. In case of a failure of ADSM/6000 server 1, you have to execute the following command on ADSM/6000 server 2 to import the data:

```
adsm> import node nodename filedata=all devclass=FS replacedefs=yes
```

After this command has completed, the ADSM/6000 client can continue operation using ADSM/6000 server 2. To control the switch of which ADSM server an ADSM client uses, edit the dsm.opt file with two servers, and use the *-servername* option on the command line, such as in:

```
dsmc query backup -servername=adsm2 /mydata
```

15.5 Standby ADSM Server

Another way to improve availability is to introduce a second ADSM server as standby server. This standby server will remain idle, but will contain some basic ADSM configurations. If the active ADSM server fails, the system manager makes the standby server active and provides the same services from the standby server.

We have explored three different methods to prepare the standby server for takeover to minimize user interruption. There are probably more ways of doing this, but our examples give you an idea about the options that are available. The issue about providing a standby server also introduces a question about the license that must be installed on each CPU where the ADSM server executes. At this time, we don′t know how IBM is handling the service of providing a second key for the standby server.

The methods to take over by the standby server use one of the following three utilities:

- Database DUMP/LOAD utilities
- Import/Export utilities
- Using Shared Disk configuration

We first discuss how to configure and prepare the standby server for each of these methods and then compare the methods.

15.5.1 ADSM/6000 Server Startup Operation

As mentioned previously, several components are important to the operation of an ADSM/6000 server. They are :

- ADSM database volumes
- ADSM log volume
- ADSM configuration files (*dsmserv.dsk* and *dsmserv.opt*)
- ADSM storage pools

To understand how a standby ADSM server can take over these components, it is important to know how the ADSM server uses these components when it starts.

1. Change the current directory to access the one where the ADSM/6000 server is installed (usually /usr/lpp/adsmserv/bin).
2. Start the ADSM/6000 server.
3. Search for the dsmserv.opt file in the directory pointed at by the DSM_SERV environment variable or in the current directory.
4. The ADSM/6000 server always searches for the dsmserv.dsk file in the current directory.
5. The dsmserv.dsk file has information about the location of the ADSM/6000 database and the recovery log.
6. The database has information about where the storage pools are located.

In our environment we are using the */var/adsmserv* directory as the configuration directory and the */usr/lpp/adsmserv/bin* directory as the installation directory. That means the ADSM/6000 will search for the *dsmserv.opt* file in the */var/adsmserv* directory and for the *dsmserv.dsk* file in the */usr/lpp/adsmserv/bin* directory.

15.5.2 Takeover with Dump/Load Utility

The first method is to use the ADSM Dump/Load utility for recovery on the standby server. During normal operation, the active ADSM/6000 server continuously updates the database and transaction log. To ensure that a copy of recent database information is available, it is recommended that the database be dumped at the following times :

- After extensive backup or archive activities have taken place from client nodes
- After migration or reclamation processes have completed
- After moving client data files from one volume to another volume

The most effective solution is to dump the database periodically to a data server, such as described in 15.2.1, “How to Dump the Database” on page 129.

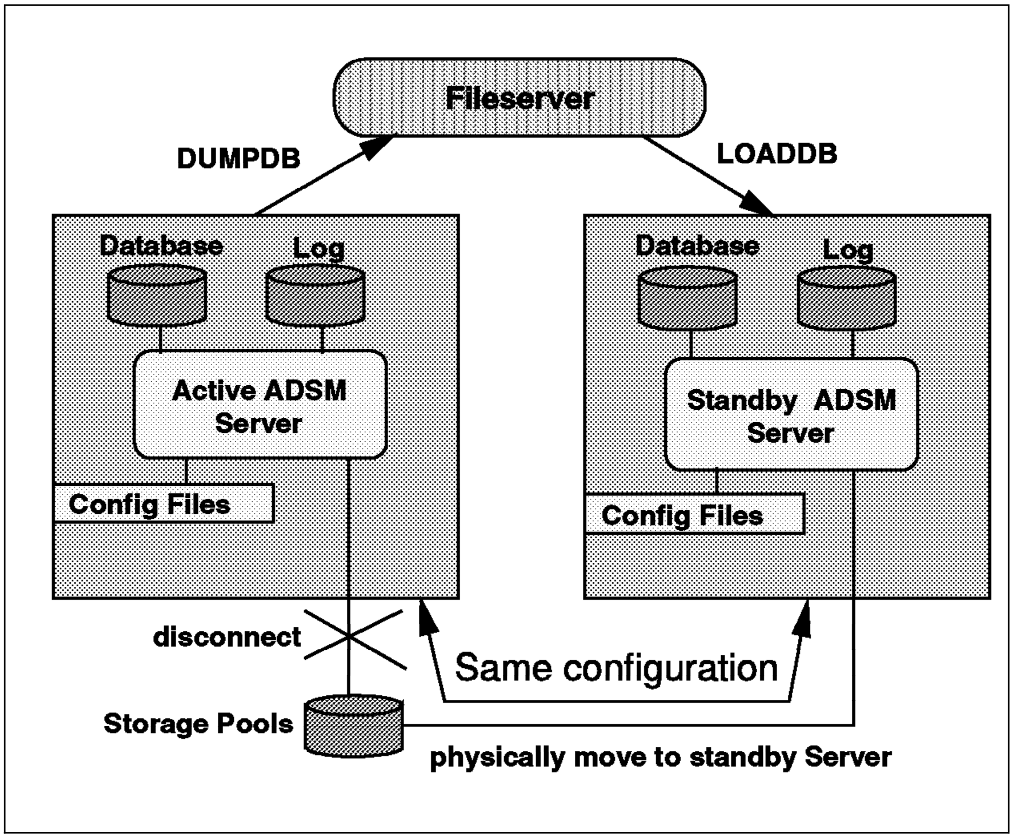


Figure 41. Takeover by DUMP/LOAD ADSM Database

15.5.2.1 Recovery Steps

The ADSM standby server must be configured with the same configuration as the active running server. This means that the database and log files are positioned at the same location and have the same size, and that the configuration files look the same. The difference to the active server is that the database and the log file are empty, and the standby server does not have access to any storage pools. We also assume that the storage pools are not broken. If they are broken, this solution is of no great value because the users cannot access the old data, but only backup/archive new data.

The following steps must be taken on the standby server to activate the ADSM/6000 server if the active ADSM/6000 server breaks:

1. Physically detach the storage pools from the broken ADSM/6000 server and attach them to the standby server. They must be accessible under the same device names. For example, if a tape unit has been accessed as /dev/mt1 on ADSM/6000 server 1, it must also be configured as /dev/mt1 on the standby server. The best situation is if the storage pools reside on a data server that is still available, and you can access the ADSM volumes from the standby server with the same path name (NFS mounted).

2. Verify that the ADSM/6000 server option file (*dsmserv.opt*) contains an entry that defines a device class and allows access to the dumped database. For example, an entry to a device class of type FILE, as in the following line.

```
DUMPloaddb Devclass dumpdb Devtype=FILE dir=/NFSshare/dir
```

3. Use the stand-alone *Load* utility to restore the dumped database from the primary server to the backup server.

```
dsmserv loaddb devclass=dumpdb volname=00000001.DMP consistency=yes
```

4. Hopefully, the dump has been taken using the *consistency=yes* option. If not, you need to audit the database and eventually fix it.

```
dsmserv auditdb fix=yes
```

15.5.3 Takeover with Export/Import Utility

The second method allows a recovery on the standby server by using the ADSM/6000 *Export/Import* utility. Although the same ADSM Export/Import utility is used to copy to removable media either an entire server or a part of it, exporting the whole ADSM/6000 server is required for disaster recovery. The export information from the primary server should include the following:

- Administrator definitions
- Client node definitions
- Policy and scheduling definitions
- Backed up and archived files
- Active backed up and archived files

Note that the storage pool definitions are not exported.

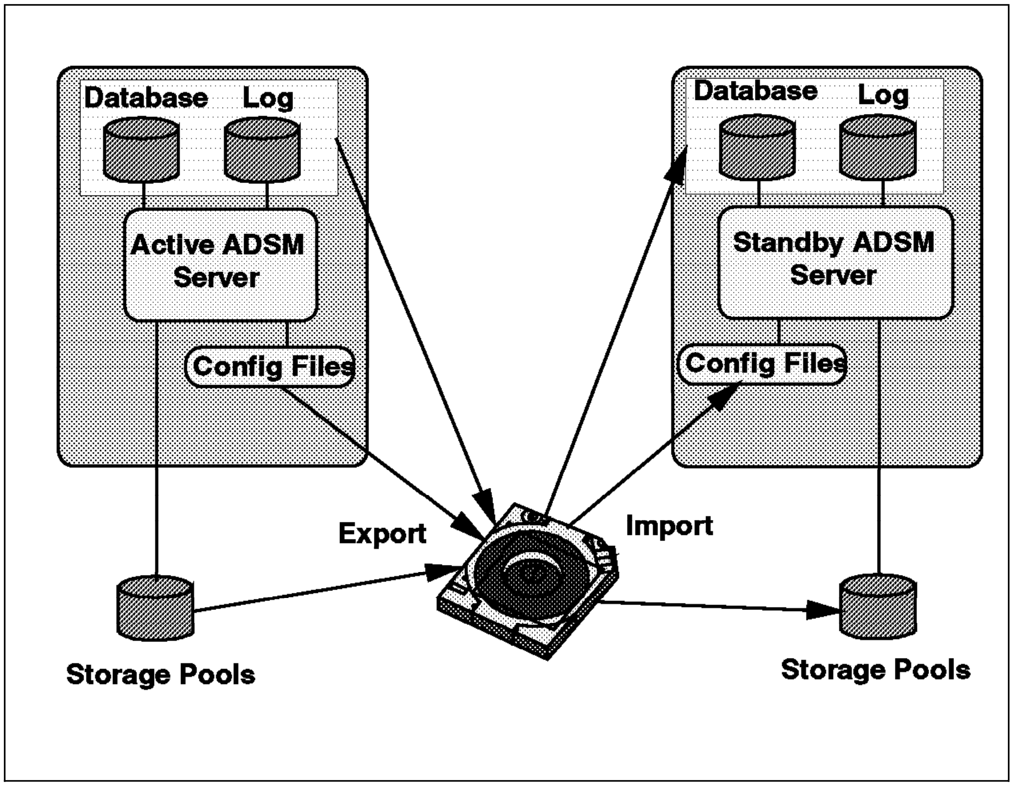


Figure 42. Takeover with Export/Import Utility

15.5.3.1 Recovery Steps

The ADSM standby server has to be configured similar to the configuration of the active running server. This means that the database and log must have a size that is large enough to hold the information when importing, but there is no need to position them at exactly the same location on the file system. Also, the storage pools have to be created in advance. However, it is probably better to create the storage pool volumes just shortly before you import. This has the advantage that the space is allocated when needed.

Use the following steps to install the standby server using the import utility.

1. Verify that the database volumes and log volumes have been defined and that enough space is allocated to load the database from ADSM/6000 server 1.

2. Verify that the storage pools and storage volumes have been defined, and that enough space has been allocated.

3. Import all database and data information, and capture error messages to a file.

   ```
   dsmadm -consolemode -output=/var/adsmserv/import.error
   adsm> import Server filedata=all Devclass=cartridge
   ```

   During the import process, ADSM/6000 renames the imported active policy set on the tape to $$ACTIVE$$. After the import, ADSM/6000 activates this policy set and validates the policy set definitions.

15.5.4 Shared Disk Configuration

The third method that allows quick recovery is to use a shared disk configuration. In this scenario, an external disk subsystem, such as the IBM 9333 DASD, is attached to two ADSM servers in a twin-tailed configuration. In the normal operation, the disk, hence all the ADSM/6000 server information and data, is owned by the primary server. The disk subsystems, although attached to the backup server, are not available for the backup server.

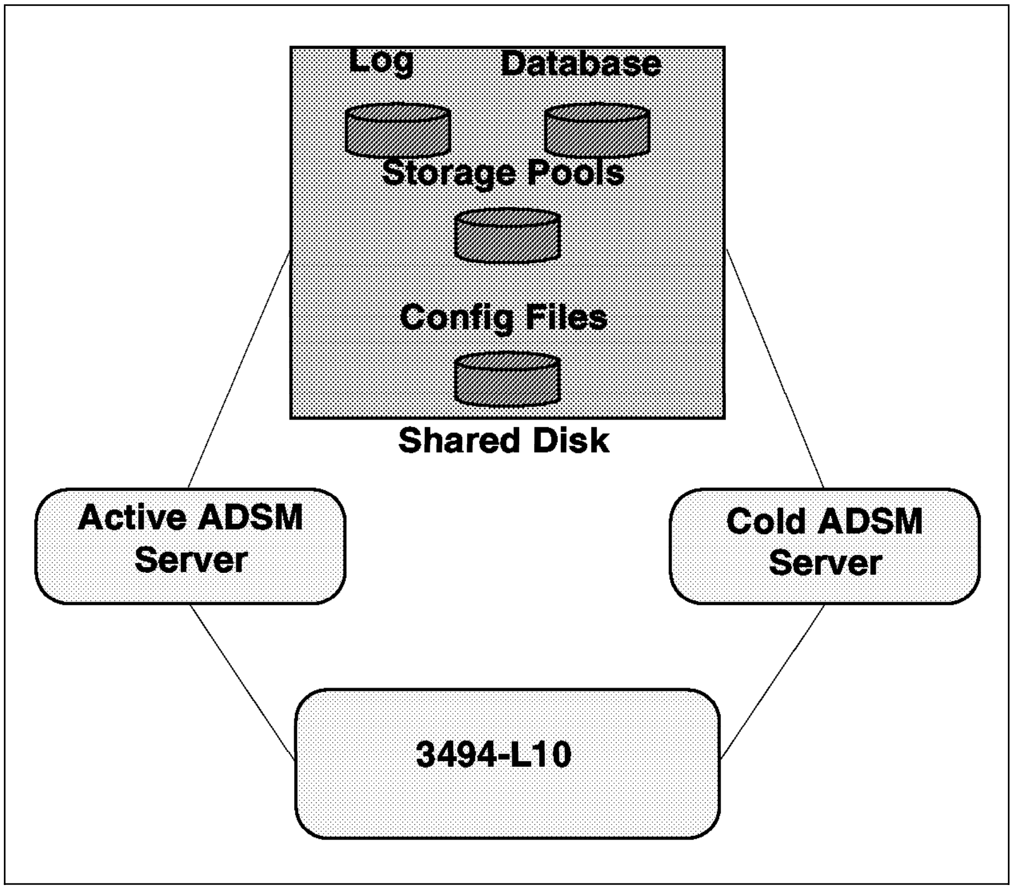


Figure 43. Shared Disk Configuration

This setup is similar to the IBM AIX High-Availability Cluster Multi-Processing/6000 (thereafter named HACMP/6000) mode 1 configuration. The primary active server should be configured in the following way:

- Create a new volume group on the external disk subsystems, for example, *adsmvg*.
- Create a new journal file system (JFS) called /var/adsmserv on the new volume group.
- Copy the ADSM server option file from /usr/lpp/adsmserv/bin to /var/adsmserv (*dsmserv.opt*) and set up a symbolic link.

```
# cp /usr/lpp/adsmserv/bin/dsmserv.opt /var/adsmserv/dsmserv.opt
# ln -s /var/adsmserv/dsmserv.opt /usr/lpp/adsmserv/bin/dsmserv.opt
```

- Copy the ADSM server configuration file from /usr/lpp/adsmserv/bin to /var/adsmserv and set up a symbolic link.

```
# cp /usr/lpp/adsmserv/bin/dsmserv.dsk /var/adsmserv/dsmserv.dsk
# ln -s /var/adsmserv/dsmserv.dsk /usr/lpp/adsmserv/bin/dsmserv.dsk
```

- Define the database volume and recovery log on the /var/adsmserv directory using the `dsmfmt` command. We do not recommend using the logical volume manager (LVM) to create the volume because some volume information will be stored under /dev directory. For example, we created the database volume /var/adsmserv/dbvol and recovery log volume /var/adsmserv/logvol by using:

```
dsmfmt -m -log /var/adsmserv/logvol 9
dsmfmt -m -db /var/adsmserv/dbvol 21
```

However, it is suggested that you define the `log` and `db` file on different filesystems to avoid that `/var` be filled up by the `log` file.

- Create all storage volumes, such as the archive volumes and backup volumes, under the /var/adsmserv directory. Use the dsmfmt command instead of LVM.
- Start the ADSM server from the directory /usr/lpp/adsmserv/bin.

```
# cd /usr/lpp/adsmserv/bin
# dsmserv
```

The primary server is operational in the normal way like any other ADSM/6000 server. In the case of failure of the primary ADSM/6000 server, you can cause the standby server to takeover the primary server by performing these steps:

1. Umount the /var/adsmserv filesystem from the primary server.
2. Varyoff the adsmvg volume group from the primary server.

```
varyoffvg adsmvg
```

3. Export the adsm volume group from the primary server.

```
exportvg adsmvg
```

4. On the standby server, import the adsm volume group.

```
importvg -y adsmvg hdiskX
```

   where hdiskX is the disk containing the adsmvg volume group.

5. Varyon the volume group on the standby server.

```
varyonvg adsmvg
```

6. Set up symbolic links for the *dsmserv.opt* and for the *dsmserv.dsk* file.

```
# ln -s /var/adsmserv/dsmserv.opt /usr/lpp/adsmserv/bin/dsmserv.opt
# ln -s /var/adsmserv/dsmserv.dsk /usr/lpp/adsmserv/bin/dsmserv.dsk
```

7. Start the ADSM server from the directory /usr/lpp/adsmserv/bin.

```
cd /usr/lpp/adsmserv/bin
dsmserv
```

If the CPU of the primary server goes down completely, steps 1-3 are not required since the volume group will have been varyed off. In this case, only *importvg* and *varyonvg* are required on the standby ADSM server.

15.5.5 Comparisons

15.5.5.1 Load/Dump Utility

- Most recent database updates may not be recoverable.
- Consumes disk space as the database volume, and the log volume may have been allocated on the standby server in advance.
- Requires you to define storage pools and data volumes after restoring the dump on the standby server.

15.5.5.2 Export/Import Utility

- Most recent database updates may not be recoverable.
- Consumes disk space as the database volume, and the log volume may have been allocated on the standby server.
- Requires you to define storage pools and data volumes after the import.
- Provides flexibility to import user specified data only.

15.5.5.3 Shared Disk Configuration

- The most recent database updates are available.
- Does not consume disk space on the backup server.
- ADSM/6000 server database definitions available on the primary server are available on the backup server once the volume group is imported and varied on to the backup server.
- Can easily integrate with HACMP/6000 if preferred.
- Takeover maneuver is the quickest and requires little configuration.

Appendix A. ADSM Configuration Files

The following is an example of the client options files.

In this example, we have two interfaces defined. The first interface uses the high-performance switch; therefore, the first IP address is one that the SP2 systems administrator has defined to the node′s switch port. The second interface is an Ethernet connection. This allows clients to operate over either network, as appropriate.

dsm.sys

```
Servername          sp2sw02
COMMmethod          TCPIP
TCPPort             1500
 TCPServeraddress   sp2sw02.itsc.pok.ibm.com
 TCPBuffsize        32
 TCPWindowsize      640 (*)
 Passwordaccess     generate
 Schedmode          prompted
 Schedlogname       /var/adsm/sched.log
 InclExcl           /var/adsm/inclexcl.list

SErvername          sp2n02
 COMMmethod         TCPIP
 TCPPort            1500
 TCPServeraddress   sp2n02.itsc.pok.ibm.com
 TCPBuffsize        32
 TCPWindowsize      24
 Passwordaccess     generate
 Schedmode          prompted
 Schedlogname       /var/adsm/sched.log
 InclExcl           /var/adsm/inclexcl.list
```

(*) The standard 24KB limit for `TCPWindowsize` is cancelled by PTF 20258. For more information, see Section 12, “Multiple ADSM/6000 Servers” on page 103.

The following is an example of the ADSM client option file and the include/exclude lists of file.

dsm.opt

```
Servername sp2sw02
Domain /appl15
tapeprompt No
```

inclexcl.list

```
include /appl15/.../* heavyclass
```

Appendix B. The adsmc Interface

The following scripts are designed to replace the command line interface of ADSM/6000. The first script checks whether the *dsmc* command is invoked in interactive or batch mode and calls the second script. The following is the listing of the first script.

```
#!/bin/ksh
#==========================================================================
#
# This script replaces the DSMC user client command of ADSM.
# It calls another script named dsmc_i and interfaces it to
# dsmc, providing proper input for both batch and interactive
# processing.
#
#==========================================================================
#
# Author: Francesco Pedulla′
# Date:   30.11.94
#
#==========================================================================
#
# Current limitations: none
#
# Known bugs: none
#
#==========================================================================
#
# Main
#
#set -x
DSMC=″/usr/lpp/adsm/bin/dsmc″
DSMC_I=″/usr/local/bin/dsmc_i″
DSMCINP=″/tmp/dsmc.input″
COMMAND=$1
typeset -u COMMAND
case $COMMAND in
  (LOOP | -PAS* | -SE*)
    $DSMC_I | $DSMC;
    ;;
  (?*)
    set -f;
    echo $* > $DSMCINP;
    set +f;
    echo ″quit″ >> $DSMCINP;
    $DSMC_I | $DSMC < $DSMCINP;
    ;;
  (*)
    $DSMC_I | $DSMC;
    ;;
esac
```

The second script acts as a filter of the user input. It performs the following logical functions:

1. Extracts from the user command all the referenced files.

2. Expands wildcards to get a list of files.

3. For each file in the list:
    a. Finds out a hard-link name for each file by replacing all symbolic links in the file name.
    b. Checks if the file is local or remote.
    c. If it is local, runs the user command for it.
    d. If it is remote, reschedules the execution of the command to the NFS server.

```
#!/bin/ksh
#==========================================================================
#
# This script replaces the DSMC user client command of ADSM.
# (for interactive use only).
# It replaces any symbolic link in the filename provided by the
# user (prepended with the current working directory) with a hard
# link name of the file. It uses that name for interfacing to ADSM.
# It also checks where the file physically resides and reschedule
# its execution on the NFS file server (if it exists).
#
#==========================================================================
#
# Author: Francesco Pedulla′
# Date:   3.11.94
#
#==========================================================================
#
# Current limitations:
#    1) does not retrieve/restore deleted files
#    2) error management is not identical to the original command′s
#    3) does not support the -PICK option
#
# Known bugs: none
#
#==========================================================================
#
# Function CD_TO_PARENT: CD to parent of filename provided as parameter
#
cd_to_parent()
{
#set -x
#
# If the path is absolute, first CD to /
#
if [[ $1 > ″.z″ && $1 < ″0″ ]]
  then
  cd /
  fi
#
# Parse the path name
#
OLD_IFS=$IFS
IFS=″/″
set $1
IFS=$OLD_IFS
#
# Step into each component
```

```
#
while [[ -d $1 ]]
  do
  cd $1
  shift
  done
export FNAME=$1
return
}
#
#===========================================================================
#
# Filename resolution routine (cleans up from symbolic links)
#
resolve_filename()
{
#
# Check if the file exists: if it does not then exit
#
#set -x
FILENAME=$1
PHYS_FILENAME=″″
FNAME=″″
if [[ ! -a $FILENAME ]]
  then
  return
  fi
#
# If the file is a directory or a symbolic link to a directory, just CD to it
#
if [[ -d $FILENAME ]]
  then
  cd $FILENAME
  FNAME=″″
#
# If the file is a symbolic link to an ordinary file, set FILENAME to point
# to it and CD to its parent directory - repeat till you hit a true directory
#
elif [[ -L $FILENAME && -f $FILENAME ]]
  then
  while [[ -L $FILENAME ]]
    do
    set ′ls -l $FILENAME′
    shift;shift
    FILENAME=$9
    cd_to_parent $FILENAME
    FILENAME=$FNAME
    done
#
# If the file is an ordinary file, set FILENAME to point
# to it and cd to its parent directory
#
elif [[ -f $FILENAME ]]
  then
  cd_to_parent $FILENAME
#
# Else, this a type of of file that we don′t handle
#
else
```

```
# echo ″The file ″ $FILENAME ″ is not an ordinary file or a directory″
# echo ″Exiting ...″
# exit -1
  return
  fi
#
# Initialize hostname (drop the domain extension)
#
OLD_IFS=$IFS
IFS=″.″
set ′hostname′
IFS=$OLD_IFS
LOCALHOST=$1
#
# Extract the mount point info from DF command output: it must be
# either a device (e.g. /dev/hd1) or a hostname (e.g. sp2cw0)
#
set ′df .′
shift;shift;shift;shift;shift;shift;shift
MOUNTED_ON_FILESYS=$9
OLD_IFS=$IFS
OLD3=$3
IFS=″:″
set $3
IFS=$OLD_IFS
MOUNTED_FROM=$1
MOUNTED_FILESYS=$2
#
# Print all relevant info for check
#
if [[ $1 != $OLD3 && $1 != $LOCALHOST ]]
  then
  LOCAL=″no″
# echo ″REMOTE FILE″
# echo ″   host ″ $MOUNTED_FROM
# echo ″   filesystem ″ $MOUNTED_FILESYS ″ mounted on ″ $MOUNTED_ON_FILESYS
else
  LOCAL=″yes″
# echo ″LOCAL FILE″
# echo ″   device ″ $MOUNTED_FROM
  fi
#
# CD up from file and follow symbolic links to find the physical filename
#
i=″0″
while [[ $PWD != ″/″ ]]
  do
#
# If the current directory is a symbolic link, move to the linked directory
# and repeat till you get into a true directory (also, check if the symbolic
# link uses relative or absolute pathnames)
#
  while [[ -L $PWD ]]
    do
    set ′ls -l $PWD′
    shift;shift
    if [[ $9 < ″.z″ || $9 > ″0″ ]]
      then
      cd ..
      fi
    cd $9
    done
#
# Parse the directory name out of $PWD and store in in DIRS[]
#
  OLD_IFS=$IFS
  IFS=″/″
  set ′pwd′
  IFS=$OLD_IFS
  while [[ -n $2 ]]
    do
    shift
    done
  export DIRS[$i]=$1
  cd ..
  let i=$i+1
  done
NUMELE=$i-1
#
# Build physical file name and print it out
#
let i=$NUMELE
while [[ $i -ge 0 ]]
  do
  PHYS_FILENAME=$PHYS_FILENAME″/″${DIRS[$i]}
  let i=$i-1
  done
#echo ″   physical file name: ″ $PHYS_FILENAME″/″$FNAME
#
# end
#
return
}
#
#==========================================================================
#
# Function set_last_options
#
set_last_options()
{
#set -x
export LAST_OPTIONS=″″
while [[ -n $1 ]]
  do
  LAST_OPTIONS=$LAST_OPTIONS″ ″$1
  shift
  done
}
#
#==========================================================================
#
build_remote_filename()
{
# For remote files, parse the mount point directory filename
# and the physical filename
#
#set -x
OLD_IFS=$IFS
```

```
IFS=″/″
set -A RDIRS $MOUNTED_ON_FILESYS
set -A LDIRS $PHYS_FILENAME
IFS=$OLD_IFS
#
# Build the filename relative to the mount point
#
REL_FILENAME=″″
i=″0″
while [[ -n ${LDIRS[$i]} ]]
  do
  if [[ ${LDIRS[$i]} != ${RDIRS[$i]} ]]
    then
    REL_FILENAME=$REL_FILENAME″/″${LDIRS[$i]}
    fi
  let i=$i+1
  done
#echo ″   relative file name: ″ $REL_FILENAME″/″$FNAME
#
# Build the file name for the NFS server and list it
#
REMOTE_FILENAME=$MOUNTED_FILESYS$REL_FILENAME
#echo ″   remote file name: ″ $REMOTE_FILENAME″/″$FNAME
return
}
#
#===========================================================================
#
# Main
#
DSMC=″/usr/lpp/adsm/bin/dsmc″
WD=$PWD
#
# Extract DSMC command
#
#set -x
read
set -f
if [[ -n $REPLY ]] then set $REPLY; fi;
set +f
while [[ $1 != ″quit″ ]]
do
COMMAND=$1
typeset -u COMMAND
case $COMMAND in
  (D* | Q* | SET)  COMMAND=$COMMAND″ ″$2;
     shift;;
esac
typeset -u COMMAND
case $COMMAND in
  (SET\ A*) export COMMAND=$COMMAND″ ″$2;
     typeset -u COMMAND;
     shift;;
esac
#
# Set options (they all start with ″-″)
#
OPTIONS=″ ″
while [[ $2 < ″.″ && $2 > ″,″ ]]
```

```
  do
  OPTIONS=$OPTIONS″ ″$2
  shift
  done
#
# Extract file list and last options
#
case $COMMAND in
  (SET\ P*) shift;;
  (A*) shift;
     set -A FILEARRAY $*;
     set -s ″″;
     ;;
  (*) FILELIST=$2;
     set -A FILEARRAY $FILELIST;
     if [[ -n $1 ]] then shift; fi;
     if [[ -n $1 ]] then shift; fi;
     ;;
esac
set_last_options $*
#
# Some DSMC command do not require filename processing; for the others,
# if multiple filenames are provided, invoke DSMC for each of them
# and process only the first one in the current script
#
case $COMMAND in
  (D*\ AC* | D*\ F* | LOOP | Q*\ AC* | Q*\ F* | Q*\ M* | Q*\ S* | SET*\ P*)
    echo $COMMAND $OPTIONS $LAST_OPTIONS
    ;;
  (H* | \?)
    echo $COMMAND
    ;;
  (A* | D*\ AR* | I* | Q*\ AR* | Q*\ B* | SEL* | SET*\ A* | RES* | RET*)
    j=″0″
    if [[ -z ${FILEARRAY[$j]} ]]
      then
      echo $COMMAND $OPTIONS $PHYS_FILENAME″/″$FNAME $LAST_OPTIONS
      fi
    while [[ -n ${FILEARRAY[$j]} ]]
      do
      if [[ $1 > ″.z″ && $1 < ″0″ ]]
        then
        FILEARRAY[$j]=$WD″/″${FILEARRAY[$j]}
        fi
      cd $WD
      resolve_filename ${FILEARRAY[$j]}
      cd $WD
      if [[ $LOCAL = ″yes″ ]]
        then
        echo $COMMAND $OPTIONS ${FILEARRAY[$j]} $LAST_OPTIONS
      else
        if [[ -n $FNAME ]]
          then
          build_remote_filename
          echo ″rsh″ $MOUNTED_FROM $DSMC $COMMAND $OPTIONS \
            $REMOTE_FILENAME″/″$FNAME $LAST_OPTIONS
        else
          echo $COMMAND $OPTIONS ${FILEARRAY[$j]} $LAST_OPTIONS
          fi
        fi
      let j=$j+1
      done
    ;;
  (*)
    echo $COMMAND $OPTIONS $FILEARRAY[0] $LAST_OPTIONS
    ;;
esac
cd $WD
read
set -f
if [[ -n $REPLY ]] then set $REPLY; fi;
set +f
done
echo quit
```

Appendix C. SP2 Users Authorization

```
#==========================================================================
#
# This script let the ADSM administrator authorize a user
# to access data from any node on the SP2 system.
#
#==========================================================================
#
# Author: Francesco Pedulla’
# Date:   17.11.94
#
#==========================================================================
#
# Current limitations:
#
# Known bugs: none
#
#==========================================================================
#
# Main
#
OPERATION=$1
shift
FILESPEC=$1
shift
NETWORK=$1
shift
OPTIONS=$*
set -A SPDATAARRAY ’/usr/lpp/ssp/bin/splstdata -a’
#
# Loop on all the nodes on the selected network
#
i=”13”
while [[ -n ${SPDATAARRAY[$i]} ]]
  do
  if [[ $NETWORK = ${SPDATAARRAY[$i]} ]]
    then
    let i=i+1
    set ’host ${SPDATAARRAY[$i]}’
    NODENAME=$1
    /usr/lpp/adsm/bin/dsmc set access $OPERATION $FILESPEC $NODENAME $OPTIONS
    let i=i+6
  else
    let i=i+7
  fi
done
#
# THE END!
#
exit 0
```

Appendix D. Device Support

The following devices are currently supported by ADSM/6000 on the IBM POWERparallel Systems SP2.

- Tape Units
  - IBM 7208-1 8mm tape drive
  - IBM 7208-11 8mm tape drive
  - EXB 8200 8mm tape drive
  - EXB 8205 8mm tape drive
  - EXB 8205XL 8mm tape drive
  - EXB 8500 8mm tape drive
  - EXB 8500C 8mm tape drive
  - EXB 8505 8mm tape drive
  - EXB 8505XL 8mm tape drive
  - Transitional Technologies, Inc. CST 8000H
  - Transitional Technologies, Inc. CST 8519H
  - IBM 3480 B11, B22 (BMPX)
  - IBM 3490E C10,C11,C1A,C22,C2A,E01,E11 (SCSI)
  - IBM 3490/3490E B02,B04,B20,B40,C10,C11,C1A,C22,C2A,D31,D32 D41,D42 (BMPX)
  - IBM 3490/3490E B20,B40,C10,C11,C1A,C22,C2A (ESCON)
- Tape libraries
  - EXB-10e 8mm tape library
  - EXB-10i 8mm tape library
  - EXB-60 8mm tape library
  - EXB-120 8mm tape library
  - EXB-210 tape library
  - Lago LS/380L 8mm tape library
  - Transitional Technologies, Inc. Series-10
  - IBM 3494-L10 Tape Library (SCSI, BMPX, ESCON)
  - IBM 3495 Tape Library (BMPX, ESCON)
- Disk Storage Devices
  - Any disk supported by AIX/6000 with LVM
    - Files through JFS
    - Raw logical volumes
- Optical Devices
  - IBM 7209-01 optical drive
  - IBM 7209-02 optical drive

  - IBM 3995 Model 063
  - IBM 3995 Model 163
  - IBM 3995 Model A63

You should be aware that ADSM provides its own device drivers for use with 8mm tape drives. These device drivers must be used in place of the device drivers that are distributed with AIX. To ensure that you use the ADSM drivers instead of the base AIX ones, be sure that you select one of the following items from the list that is presented to you by SMIT after you have chosen the "Devices," "Tape Drive," "Add a Tape Drive" sequence:

- IBM_7208_1
- IBM_7208_11
- EXB_8200
- EXB_8500
- EXB_8500C

Note that you should *not* select the "8mm5gb," "8mm" or "ost" items, since they are the base AIX 8mm tape device drivers.

List of Abbreviations

ACL	access control lists
ADSM	ADSTAR Distributed Storage Manager
ADSTAR	Advanced Storage And Retrieval
AIX	advanced interactive executive (IBM′s flavor of UNIX)
Amd	BSD automounter daemon *This product includes software developed by The University of California, Berkeley and its Contributors.*
API	application program interface
BOS	base operating system
CAD	computer aided design
CD	change directory (DOS , OS/2, and AIX command)
CD-ROM	(optically read) compact disk - read only memory
CMI	corporate manufacturing instruction
CPU	central processing unit
CSL	cartridge stack loader
CSS	communication subsystem (IBM AIX PSSP component)
DSM	distributed systems management
FDDI	fiber distributed data interface
FIFO	first in/first out
FTP	file transfer protocol
GB	gigabyte
GUI	graphical user interface
HPS	high-performance switch
I/O	input/output
IBM	International Business Machines Corporation
ID	identification/identifier
IP	internet protocol (ISO)
IPL	initial program load
ITSO	International Technical Support Organization
JFS	journaled file system (AIX)
KB	kilobyte
LVM	logical volume manager
MB	megabyte
NFS	network file system (USA, Sun Microsystems Inc)
NTP	network time protocol
ODM	object data manager (AIX)
POWER	performance optimization with enhanced RISC
PSSP	AIX Parallel System Support Programs
PTF	program temporary fix
RAID	Redundant Array of Independent Disks
RISC	reduced instruction set computer/cycles
ROM	read only memory
SCSI	small computer system interface
SDR	system data repository
SMIT	System Management Interface Tool (see also DSMIT)
SP	Scalable Power parallel
TCP	transmission control protocol (USA, DoD)
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	user datagram protocol (TCPIP)
UNIX	an operating system developed at Bell Laboratories
USA	United States of America
VNET	virtual node exchange transmission

Index

Printed in U.S.A.

GG24-4499-00

Artwork Definitions

id	File	Page	References
ITSLOGO	4499SU	i	i

Table Definitions

id	File	Page	References
R1	AH3ADAE	70	71
R2	AH3ADAE	70	71
R1X	AH3ADBG	86	87
R2X	AH3ADBG	86	87

Figures

id	File	Page	References
AADAF1	AH3AADA	5	1
H3AADC	AH3AADC	7	2 6
AAC1	AH2AAA	10	3 9, 27
AAAC1	AH3AAAC	12	4 11
AAAABF1	AH3AAAA	14	5 13
AAAABF2	AH3AAAA	15	6 15
AAAACF3	AH3AAAA	16	7
AABAF1	AH3AABA	20	8 19
PLAN4	AH2AAC	28	9 28
ACBAF1	AH3ACBA	44	10 44
ACBAF2	AH3ACBA	44	11 44
ACBAF3	AH3ACBA	45	12 45
ACBAF4	AH3ACBA	45	13 45
ACBBF1	AH3ACBB	47	14 47
ACBBF2	AH3ACBB	47	15 47
ACBBF3	AH3ACBB	48	16 48
ACBBF4	AH3ACBB	50	17 50
ACBBF5	AH3ACBB	51	18 50
ACBCF1	AH3ACBC	54	19
ACBCF2	AH3ACBC		

id	File	Page	References
		55	20 54
ACCAF1	AH3ACCA	58	21 58
ACCBF1	AH3ACCB	59	22 59
ACCBF2	AH3ACCB	60	23
STSCEN	AH2ADA	65	24 65
ADSM3	AH3ADAE	70	25 69
COMSCEN	AH2ADB	74	26 73
ADSM1	AH3ADBF	81	27
PD1	AH3ADBF	82	28
PD2	AH3ADBF	82	29
ADSMS2	AH3ADBG	84	30
PD3	AH3ADBG	85	31
MULT0	AH2AED	105	32 103, 104
MULT1	AH2AED	107	33 106
MULT2	AH2AED	108	34 107
MULT3	AH2AED	109	35 108
SYMB	AH3AFAA	118	36
VIRT1	AH3AFAB	122	37
VIRT2	AH3AFAB	123	38 123
EXPORT1	AH2AGE	134	39
EXPORT2	AH2AGE	135	40
STAND1	AH3AGFB	138	41
STAND2	AH3AGFC	139	42
STAND3	AH3AGFD	141	43

Headings

id	File	Page	References
NOTICES	4499FM	xiii	Special Notices ii
BIBL	4499PREF	xv	Related Publications
H0AA	AH1AA	1	Chapter 1, Introduction xv
H1AAD	AH2AAD	3	Section 1, Storage Management Overview
H2AADB	AH3AADB	3	1.1, Hierarchical Storage Management
H2AADA	AH3AADA	4	1.2, Multilevel Hierarchy
H3AADC	AH3AADC	5	1.2.1, Storage Devices
H4AADCA	AH3AADC	5	1.2.1.1, Disk Storage
H4AADCB	AH3AADC	6	1.2.1.2, Tape Storage

H4AADCC	AH3AADC	6	1.2.1.3, Optical Storage
H1AAA	AH2AAA	9	Section 2, ADSM Overview
H2AAAC	AH3AAAC	9	2.1, ADSM Vocabulary
H3AAACA	AH3AAAC	9	2.1.1, Storage Pool
H3AAACB	AH3AAAC	9	2.1.2, Storage Management Policies
H3AAACC	AH3AAAC	12	2.1.3, Central Scheduler
H3AAACD	AH3AAAC	13	2.1.4, Nodes
H2AAAA	AH3AAAA	13	2.2, Components of ADSM/6000
H3AAAAA	AH3AAAA	13	2.2.1, The ADSM Server
H3AAAAB	AH3AAAA	15	2.2.2, Administrative Client
H3AAAAC	AH3AAAA	16	2.2.3, Backup/Archive ADSM Client
H3AAAAD	AH3AAAA	17	2.2.4, Application Programming Interface (API)
H2AAAB	AH3AAAB	17	2.3, ADSM Main Functions
H3AAABA	AH3AAAB	17	2.3.1, Backup and Restore
H3AAABB	AH3AAAB	18	2.3.2, Archive and Retrieve
H1AAB	AH2AAB	19	Section 3, SP2 Overview
H2AABA	AH3AABA	19	3.1, Hardware
H3AABAA	AH3AABA	20	3.1.1, Nodes
H3AABAB	AH3AABA	21	3.1.2, High-Performance Switch (HPS)
H3AABAC	AH3AABA	21	3.1.3, Control Workstation
H3AABAD	AH3AABA	21	3.1.4, SP2 Ethernet
H3AABAE	AH3AABA	21	3.1.5, Serial Lines
H2AABB	AH3AABB	22	3.2, SP2 Software 21
H2AACB	AH3AABB	24	3.2.1, SP2 Software Environment
H1AAC	AH2AAC	27	Section 4, ADSM on SP2
H2AACA	AH2AAC	27	4.1, Roles of ADSM/6000
H2AACAX	AH2AAC	27	4.2, Data on the IBM POWERparallel Systems SP2
H2AC$B	AH2AAC	29	4.3, Selecting Storage Devices
H0AC	AH1AC	31	Chapter 2, ADSM Installation and Basic Customization xv
H1AC#	AH2AC#	33	Section 5, ADSM Customization Steps
H2AC#A	AH2AC#	33	5.1, Steps for the ADSM Server Customization
H2AC#B	AH2AC#	33	5.2, Step for the ADSM Client Customization
H1ACA	AH2ACA	35	Section 6, ADSM/6000 Installation 33
H2ACAA	AH3ACAA	35	6.1, ADSM/6000 Server Installation
H3ACAAA	AH3ACAA	35	6.1.1, Installation on a Full-AIX Node 35, 103
H3ACAAB	AH3ACAA	36	6.1.2, Installation for /usr Clients 35
H2ACAB	AH3ACAB	38	6.2, ADSM/6000 Client Installation
H3AACBA	AH3ACAB	38	6.2.1, Installation on Full-AIX Node
H3ACAX	AH3ACAB	38	6.2.1.1, Installation Procedure 39
H3AACAB	AH3ACAB		

		92	11.2, Centralized Management of Logs	
H2AEBD	AH3AEBD	94	11.3, Using the IBM 3490E Model E11	51
H2AEBE	AH3AEBE	97	11.4, Keeping the Scheduler Alive	
H2AEBF	AH3AEBF	98	11.5, Using ADSM Macros	53
H2AEBG	AH3AEBG	99	11.6, ADSM/6000 Accounting	
H2AEBH	AH3AEBH	99	11.7, Handling the ADSM Activity Log	
H2AEBI	AH3AEBI	101	11.8, ADSM Scheduler Event Log	
H2AED	AH2AED	103	Section 12, Multiple ADSM/6000 Servers	145
H3AEDA	AH2AED	103	12.1, Installing Multiple ADSM/6000 Servers	
H3AEDB	AH2AED	105	12.2, Using Multiple ADSM/6000 Servers	
H3AEDC	AH2AED	106	12.3, Planning for Multiple ADSM/6000 Servers	
H1AEC	AH2AEC	111	Section 13, Performance	
H2AECA	AH3AECA	111	13.1, Tuning Performance for the HPS	106
H3AECAA	AH3AECA	111	13.1.1, ADSM Communication	
H3AECAB	AH3AECA	112	13.1.2, TCP/IP Communication	
H3AECAC	AH3AECA	112	13.1.3, IP Interface to CSS	
H3AECAD	AH3AECA	112	13.1.4, HPS Device Driver	
H0AF	AH1AF	115	Chapter 5, User Access Issues	xv
H1AFA	AH2AFA	117	Section 14, File Names	115
H2AFAA	AH3AFAA	117	14.1, Symbolic Links	
H3AFAAB	AH3AFAA	118	14.1.1, User′s Point of View	
H3AFAAC	AH3AFAA	119	14.1.2, ADSM Usage to Bypass the Symbolic Links	
H2AFAB	AH3AFAB	119	14.2, NFS Access	
H3AFABA	AH3AFAB	119	14.2.1, NFS Files Identity	
H3AFABB	AH3AFAB	119	14.2.2, NFS Performance	
H3AFABC	AH3AFAB	120	14.2.3, ADSM Configurations to Solve NFS Problems	
H3AFAB$	AH3AFAB	121	14.2.4, Example of Use of Virtual Nodes	
H3AFABD	AH3AFAB	123	14.2.5, ADSM Client Interface	
H0AG	AH1AG	125	Chapter 6, ADSM Availability	xv
H2AGA	AH2AGA	127	Section 15, ADSM Availability Issues	14
H2AGB	AH2AGB	128	15.1, Mirror the Database and Log	
H2AGC	AH2AGC	129	15.2, Dump/Load the ADSM Database	
H2AGCA	AH2AGC	129	15.2.1, How to Dump the Database	137
H3AGCB	AH2AGC	131	15.2.2, How to Recover the Database	132
H2AGD	AH2AGD	132	15.3, ADSM Auditing	
H2AGDA	AH2AGD	132	15.3.1, Audit the ADSM Database	130, 131
H2AGDB	AH2AGD	132	15.3.2, Audit the Storage Volumes	132

H2AGDC	AH2AGD	133	15.3.3, Audit the Library
H2AGDD	AH2AGD	133	15.3.4, Audit the License
H2AGE	AH2AGE	133	15.4, Export/Import Functions
H2AGEA	AH2AGE	134	15.4.1, Example One for Export/Import
H2AGEB	AH2AGE	135	15.4.2, Example Two for Export/Import
H2AGF	AH2AGF	136	15.5, Standby ADSM Server
H3AGFA	AH2AGF	137	15.5.1, ADSM/6000 Server Startup Operation
H3AGFB	AH3AGFB	137	15.5.2, Takeover with Dump/Load Utility
H4AGFAA	AH3AGFB	138	15.5.2.1, Recovery Steps
H3AGFC	AH3AGFC	139	15.5.3, Takeover with Export/Import Utility
H4AGFCA	AH3AGFC	140	15.5.3.1, Recovery Steps
H3AGFD	AH3AGFD	140	15.5.4, Shared Disk Configuration
H3AGFE	AH3AGFE	143	15.5.5, Comparisons
H4AGFEA	AH3AGFE	143	15.5.5.1, Load/Dump Utility
H4AGFEB	AH3AGFE	143	15.5.5.2, Export/Import Utility
H4AGFEC	AH3AGFE	143	15.5.5.3, Shared Disk Configuration
H1AP1	AH1AP1	145	Appendix A, ADSM Configuration Files xv
H1AP2A	AH1AP2	147	Appendix B, The adsmc Interface xv, 123
H1AP2C	AH1AP2	155	Appendix C, SP2 Users Authorization xv, 91
H1AP3	AH1AP3	157	Appendix D, Device Support xv, 29

Index Entries

id	File	Page	References
I1AA	AH1AA	1	(1) Introduction 3, 9, 19, 27
I1AAD	AH2AAD	3	(1) Storage Management Overview 3, 4
I2AAD	AH2AAD	3	(1) Introduction (2) Storage Management Overview
I1AADB	AH3AADB	3	(1) Hierarchical Storage Management
I2AADB	AH3AADB	3	(1) Storage Management Overview (2) Hierarchical Storage Management
I1AADA	AH3AADA	4	(1) Multilevel Hierarchy 5
I2AADA	AH3AADA	4	(1) Storage Management Overview (2) Multilevel Hierarchy
I1AADC	AH3AADC	5	(1) Storage Devices 5, 6, 6
I2AADC	AH3AADC	5	(1) Multilevel Hierarchy (2) Storage Devices
I1AADCA	AH3AADC	5	(1) Disk Storage
I2AADCA	AH3AADC	5	(1) Storage Devices (2) Disk Storage
I1AADCB	AH3AADC	6	(1) Tape Storage
I2AADCB	AH3AADC		

```
                          6       (1) Storage Devices
                                  (2) Tape Storage
I1AADCC     AH3AADC
                          6       (1) Optical Storage
I2AADCC     AH3AADC
                          6       (1) Storage Devices
                                  (2) Optical Storage
I1AAA       AH2AAA
                          9       (1) ADSM Overview
                                       9, 13, 17
I2AAA       AH2AAA
                          9       (1) Introduction
                                  (2) ADSM Overview
I1AAAC      AH3AAAC
                          9       (1) ADSM Vocabulary
                                       9, 9, 12, 13
I2AAAC      AH3AAAC
                          9       (1) ADSM Overview
                                  (2) ADSM Vocabulary
I1AAACA     AH3AAAC
                          9       (1) Storage Pool
I2AAACA     AH3AAAC
                          9       (1) ADSM Vocabulary
                                  (2) Storage Pool
I1AAACB     AH3AAAC
                          9       (1) Storage Management Policies
I2AAACB     AH3AAAC
                          9       (1) ADSM Vocabulary
                                  (2) Storage Management Policies
I1AAACC     AH3AAAC
                          12      (1) Central Scheduler
I2AAACC     AH3AAAC
                          12      (1) ADSM Vocabulary
                                  (2) Central Scheduler
I1AAACD     AH3AAAC
                          13      (1) Nodes
I2AAACD     AH3AAAC
                          13      (1) ADSM Vocabulary
                                  (2) Nodes
I1AAAA      AH3AAAA
                          13      (1) Components of ADSM/6000
                                       13, 15, 16, 17
I2AAAA      AH3AAAA
                          13      (1) ADSM Overview
                                  (2) Components of ADSM/6000
I1AAAAA     AH3AAAA
                          13      (1) The ADSM Server
I2AAAAA     AH3AAAA
                          13      (1) Components of ADSM/6000
                                  (2) The ADSM Server
I1AAAAB     AH3AAAA
                          15      (1) Administrative Client
I2AAAAB     AH3AAAA
                          15      (1) Components of ADSM/6000
                                  (2) Administrative Client
I1AAAAC     AH3AAAA
                          16      (1) Backup/Archive ADSM Client
I2AAAAC     AH3AAAA
                          16      (1) Components of ADSM/6000
                                  (2) Backup/Archive ADSM Client
I1AAAAD     AH3AAAA
                          17      (1) Application Programming Interface (API)
I2AAAAD     AH3AAAA
                          17      (1) Components of ADSM/6000
                                  (2) Application Programming Interface (API)
I1AAAB      AH3AAAB
                          17      (1) ADSM Main Functions
                                       17, 18
I2AAAB      AH3AAAB
                          17      (1) ADSM Overview
                                  (2) ADSM Main Functions
I1AAABA     AH3AAAB
                          17      (1) Backup and Restore
I2AAABA     AH3AAAB
                          17      (1) ADSM Main Functions
                                  (2) Backup and Restore
I1AAABB     AH3AAAB
                          18      (1) Archive and Retrieve
I2AAABB     AH3AAAB
                          18      (1) ADSM Main Functions
                                  (2) Archive and Retrieve
I1AAB       AH2AAB
                          19      (1) SP2 Overview
                                       19, 22, 24
I2AAB       AH2AAB
```

```
                                      19       (1)  Introduction
                                               (2)  SP2 Overview
I1AABA          AH3AABA
                                      19       (1)  Hardware
                                                     20, 21, 21, 21, 21
I2AABA          AH3AABA
                                      19       (1)  SP2 Overview
                                               (2)  Hardware
I1AABAA         AH3AABA
                                      20       (1)  Nodes
I2AABAA         AH3AABA
                                      20       (1)  Hardware
                                               (2)  Nodes
I1AABAB         AH3AABA
                                      21       (1)  High-Performance Switch (HPS)
I2AABAB         AH3AABA
                                      21       (1)  Hardware
                                               (2)  High-Performance Switch (HPS)
I1AABAC         AH3AABA
                                      21       (1)  Control Workstation
I2AABAC         AH3AABA
                                      21       (1)  Hardware
                                               (2)  Control Workstation
I1AABAD         AH3AABA
                                      21       (1)  SP2 Ethernet
I2AABAD         AH3AABA
                                      21       (1)  Hardware
                                               (2)  SP2 Ethernet
I1AABAE         AH3AABA
                                      21       (1)  Serial Lines
I2AABAE         AH3AABA
                                      21       (1)  Hardware
                                               (2)  Serial Lines
I1AABB          AH3AABB
                                      22       (1)  SP2 Software
I2AABB          AH3AABB
                                      22       (1)  SP2 Overview
                                               (2)  SP2 Software
I1AACB          AH3AABB
                                      24       (1)  SP2 Software Environment
I2AACB          AH3AABB
                                      24       (1)  SP2 Overview
                                               (2)  SP2 Software Environment
I1AAC           AH2AAC
                                      27       (1)  ADSM on SP2
                                                     27, 27, 29
I2AAC           AH2AAC
                                      27       (1)  Introduction
                                               (2)  ADSM on SP2
I1AACA          AH2AAC
                                      27       (1)  Roles of ADSM/6000
I2AACA          AH2AAC
                                      27       (1)  ADSM on SP2
                                               (2)  Roles of ADSM/6000
I1AACAX         AH2AAC
                                      27       (1)  Data on the IBM POWERparallel Systems SP2
I2AACAX         AH2AAC
                                      27       (1)  ADSM on SP2
                                               (2)  Data on the IBM POWERparallel Systems SP2
I1AC$B          AH2AAC
                                      29       (1)  Selecting Storage Devices
I2AC$B          AH2AAC
                                      29       (1)  ADSM on SP2
                                               (2)  Selecting Storage Devices
I1AC            AH1AC
                                      31       (1)  ADSM Installation and Basic Customization
                                                     33, 35, 41, 57
I1AC#           AH2AC#
                                      33       (1)  ADSM Customization Steps
                                                     33, 33
I2AC#           AH2AC#
                                      33       (1)  ADSM Installation and Basic Customization
                                               (2)  ADSM Customization Steps
I1AC#A          AH2AC#
                                      33       (1)  Steps for the ADSM Server Customization
I2AC#A          AH2AC#
                                      33       (1)  ADSM Customization Steps
                                               (2)  Steps for the ADSM Server Customization
I1AC#B          AH2AC#
                                      33       (1)  Step for the ADSM Client Customization
I2AC#B          AH2AC#
                                      33       (1)  ADSM Customization Steps
                                               (2)  Step for the ADSM Client Customization
I1ACA           AH2ACA
                                      35       (1)  ADSM/6000 Installation
```

			35, 38
I2ACA	AH2ACA	35	(1) ADSM Installation and Basic Customization (2) ADSM/6000 Installation
I1ACAA	AH3ACAA	35	(1) ADSM/6000 Server Installation 35, 36
I2ACAA	AH3ACAA	35	(1) ADSM/6000 Installation (2) ADSM/6000 Server Installation
I1ACAAA	AH3ACAA	35	(1) Installation on a Full-AIX Node
I2ACAAA	AH3ACAA	35	(1) ADSM/6000 Server Installation (2) Installation on a Full-AIX Node
I1ACAAB	AH3ACAA	36	(1) Installation for /usr Clients
I2ACAAB	AH3ACAA	36	(1) ADSM/6000 Server Installation (2) Installation for /usr Clients
I1ACAB	AH3ACAB	38	(1) ADSM/6000 Client Installation 38, 39
I2ACAB	AH3ACAB	38	(1) ADSM/6000 Installation (2) ADSM/6000 Client Installation
I1AACBA	AH3ACAB	38	(1) Installation on Full-AIX Node
I2AACBA	AH3ACAB	38	(1) ADSM/6000 Client Installation (2) Installation on Full-AIX Node
I1AACAB	AH3ACAB	39	(1) Installation for /usr Clients
I2AACAB	AH3ACAB	39	(1) ADSM/6000 Client Installation (2) Installation for /usr Clients
I1ACB	AH2ACB	41	(1) ADSM Basic Customization 41, 45, 51
I2ACB	AH2ACB	41	(1) ADSM Installation and Basic Customization (2) ADSM Basic Customization
I1ACBA	AH3ACBA	41	(1) ADSM Database and Log 41, 42, 43, 43
I2ACBA	AH3ACBA	41	(1) ADSM Basic Customization (2) ADSM Database and Log
I1ACBAA	AH3ACBA	41	(1) Loading the Kernel Extension
I2ACBAA	AH3ACBA	41	(1) ADSM Database and Log (2) Loading the Kernel Extension
I1ACBAB	AH3ACBA	42	(1) Estimating Database and Log Space
I2ACBAB	AH3ACBA	42	(1) ADSM Database and Log (2) Estimating Database and Log Space
I1ACBAC	AH3ACBA	43	(1) Defining AIX Logical Volumes
I2ACBAC	AH3ACBA	43	(1) ADSM Database and Log (2) Defining AIX Logical Volumes
I1ACBAD	AH3ACBA	43	(1) Allocation of Space for Database and Log
I2ACBAD	AH3ACBA	43	(1) ADSM Database and Log (2) Allocation of Space for Database and Log
I1ACBB	AH3ACBB	45	(1) Storage Pool Environment 46, 48
I2ACBB	AH3ACBB	45	(1) ADSM Basic Customization (2) Storage Pool Environment
I1ACBBA	AH3ACBB	46	(1) Storage Pool on Disk
I2ACBBA	AH3ACBB	46	(1) Storage Pool Environment (2) Storage Pool on Disk
I1ACBBB	AH3ACBB	48	(1) Storage Pool on Tape
I2ACBBB	AH3ACBB	48	(1) Storage Pool Environment (2) Storage Pool on Tape
I1ACBC	AH3ACBC		

```
                             51        (1) Basic Administration Tasks
                                              52, 52, 52, 53, 57, 58
I2ACBC        AH3ACBC
                             51        (1) ADSM Basic Customization
                                       (2) Basic Administration Tasks
I1ACBCA       AH3ACBC
                             52        (1) Register a New Administrator
I2ACBCA       AH3ACBC
                             52        (1) Basic Administration Tasks
                                       (2) Register a New Administrator
I1ACBCB       AH3ACBC
                             52        (1) Naming the Server
I2ACBCB       AH3ACBC
                             52        (1) Basic Administration Tasks
                                       (2) Naming the Server
I1ACBCD       AH3ACBC
                             52        (1) License Registration
I2ACBCD       AH3ACBC
                             52        (1) Basic Administration Tasks
                                       (2) License Registration
I1ACBCC       AH3ACBC
                             53        (1) Nodes Registration
I2ACBCC       AH3ACBC
                             53        (1) Basic Administration Tasks
                                       (2) Nodes Registration
I1ACC         AH2ACC
                             57        (1) ADSM Verifications
I2ACC         AH2ACC
                             57        (1) ADSM Installation and Basic Customization
                                       (2) ADSM Verifications
I1ACCA        AH3ACCA
                             57        (1) Communication Verification
I2ACCA        AH3ACCA
                             57        (1) Basic Administration Tasks
                                       (2) Communication Verification
I1ACCB        AH3ACCB
                             58        (1) Storage Pool Verification
I2ACCB        AH3ACCB
                             58        (1) Basic Administration Tasks
                                       (2) Storage Pool Verification
I1AD          AH1AD
                             63        (1) Scenarios
                                              65, 73
I1ADA         AH2ADA
                             65        (1) Scientific and Technical Scenario
                                              65, 67, 68, 69, 69
I2ADA         AH2ADA
                             65        (1) Scenarios
                                       (2) Scientific and Technical Scenario
I1ADAA        AH3ADAA
                             65        (1) User Requirements
                                              66, 66
I2ADAA        AH3ADAA
                             65        (1) Scientific and Technical Scenario
                                       (2) User Requirements
I1ADAAA       AH3ADAA
                             66        (1) Programmers
I2ADAAA       AH3ADAA
                             66        (1) User Requirements
                                       (2) Programmers
I1ADAAB       AH3ADAA
                             66        (1) End Users
I2ADAAB       AH3ADAA
                             66        (1) User Requirements
                                       (2) End Users
I1ADAB        AH3ADAB
                             67        (1) NFS and JFS Data
                                              67, 67, 68
I2ADAB        AH3ADAB
                             67        (1) Scientific and Technical Scenario
                                       (2) NFS and JFS Data
I1ADABA       AH3ADAB
                             67        (1) Programmers
I2ADABA       AH3ADAB
                             67        (1) NFS and JFS Data
                                       (2) Programmers
I1ADABB       AH3ADAB
                             67        (1) End Users
I2ADABB       AH3ADAB
                             67        (1) NFS and JFS Data
                                       (2) End Users
I1ADABC       AH3ADAB
                             68        (1) Simulation
I2ADABC       AH3ADAB
                             68        (1) NFS and JFS Data
```

			(2) Simulation
I1ADAC	AH3ADAC	68	(1) NFS Data Backup
I2ADAC	AH3ADAC	68	(1) Scientific and Technical Scenario (2) NFS Data Backup
I1ADAD	AH3ADAD	69	(1) JFS Data Backup
I2ADAD	AH3ADAD	69	(1) Scientific and Technical Scenario (2) JFS Data Backup
I1ADAE	AH3ADAE	69	(1) ADSM Server Setup
I2ADAE	AH3ADAE	69	(1) Scientific and Technical Scenario (2) ADSM Server Setup
I1ADB	AH2ADB	73	(1) Commercial Scenario 75, 76, 78, 79, 80, 83
I2ADB	AH2ADB	73	(1) Scenarios (2) Commercial Scenario
I1ADBB	AH3ADBB	75	(1) Business Application Backup/Archive Setup
I2ADBB	AH3ADBB	75	(1) Commercial Scenario (2) Business Application Backup/Archive Setup
I1ADBC	AH3ADBC	76	(1) CAD Application Backup/Archive Setup
I2ADBC	AH3ADBC	76	(1) Commercial Scenario (2) CAD Application Backup/Archive Setup
I1ADBD	AH3ADBD	78	(1) Publishing Application Backup/Archive Setup
I2ADBD	AH3ADBD	78	(1) Commercial Scenario (2) Publishing Application Backup/Archive Setup
I1ADBE	AH3ADBE	79	(1) Office Application Backup/Archive Setup
I2ADBE	AH3ADBE	79	(1) Commercial Scenario (2) Office Application Backup/Archive Setup
I1ADBF	AH3ADBF	80	(1) ADSM Server 1 Setup
I2ADBF	AH3ADBF	80	(1) Commercial Scenario (2) ADSM Server 1 Setup
I1ADBG	AH3ADBG	83	(1) ADSM Server 2 Setup
I2ADBG	AH3ADBG	83	(1) Commercial Scenario (2) ADSM Server 2 Setup
I1AE	AH1AE	89	(1) Administration and Operations 91, 111
I1AEB	AH2AEB	91	(1) System Management 91, 92, 94, 97, 98, 99, 99, 101, 103
I2AEB	AH2AEB	91	(1) Administration and Operations (2) System Management
I1AEBA	AH3AEBA	91	(1) SP2 Users Definition to ADSM
I2AEBA	AH3AEBA	91	(1) System Management (2) SP2 Users Definition to ADSM
I1AEBC	AH3AEBC	92	(1) Centralized Management of Logs
I2AEBC	AH3AEBC	92	(1) System Management (2) Centralized Management of Logs
I1AEBD	AH3AEBD	94	(1) Using the IBM 3490E Model E11
I2AEBD	AH3AEBD	94	(1) System Management (2) Using the IBM 3490E Model E11
I1AEBE	AH3AEBE	97	(1) Keeping the Scheduler Alive
I2AEBE	AH3AEBE	97	(1) System Management (2) Keeping the Scheduler Alive
I1AEBF	AH3AEBF	98	(1) Using ADSM Macros
I2AEBF	AH3AEBF	98	(1) System Management

```
                                      (2) Using ADSM Macros
I1AEBG       AH3AEBG
                         99           (1) ADSM/6000 Accounting
I2AEBG       AH3AEBG
                         99           (1) System Management
                                      (2) ADSM/6000 Accounting
I1AEBH       AH3AEBH
                         99           (1) Handling the ADSM Activity Log
I2AEBH       AH3AEBH
                         99           (1) System Management
                                      (2) Handling the ADSM Activity Log
I1AEBI       AH3AEBI
                         101          (1) ADSM Scheduler Event Log
I2AEBI       AH3AEBI
                         101          (1) System Management
                                      (2) ADSM Scheduler Event Log
I1AED        AH2AED
                         103          (1) Multiple ADSM/6000 Servers
                                          103, 105, 106
I2AED        AH2AED
                         103          (1) System Management
                                      (2) Multiple ADSM/6000 Servers
I1AEDA       AH2AED
                         103          (1) Installing Multiple ADSM/6000 Servers
I2AEDA       AH2AED
                         103          (1) Multiple ADSM/6000 Servers
                                      (2) Installing Multiple ADSM/6000 Servers
I1AEDB       AH2AED
                         105          (1) Using Multiple ADSM/6000 Servers
I2AEDB       AH2AED
                         105          (1) Multiple ADSM/6000 Servers
                                      (2) Using Multiple ADSM/6000 Servers
I1AEDC       AH2AED
                         106          (1) Planning for Multiple ADSM/6000 Servers
I2AEDC       AH2AED
                         106          (1) Multiple ADSM/6000 Servers
                                      (2) Planning for Multiple ADSM/6000 Servers
I1AEC        AH2AEC
                         111          (1) Performance
                                          111
I2AEC        AH2AEC
                         111          (1) Administration and Operations
                                      (2) Performance
I1AECA       AH3AECA
                         111          (1) Tuning Performance for the HPS
                                          111, 112, 112, 112
I2AECA       AH3AECA
                         111          (1) Performance
                                      (2) Tuning Performance for the HPS
I1AECAA      AH3AECA
                         111          (1) ADSM Communication
I2AECAA      AH3AECA
                         111          (1) Tuning Performance for the HPS
                                      (2) ADSM Communication
I1AECAB      AH3AECA
                         112          (1) TCP/IP Communication
I2AECAB      AH3AECA
                         112          (1) Tuning Performance for the HPS
                                      (2) TCP/IP Communication
I1AECAC      AH3AECA
                         112          (1) IP Interface to CSS
I2AECAC      AH3AECA
                         112          (1) Tuning Performance for the HPS
                                      (2) IP Interface to CSS
I1AECAD      AH3AECA
                         112          (1) HPS Device Driver
I2AECAD      AH3AECA
                         112          (1) Tuning Performance for the HPS
                                      (2) HPS Device Driver
I1AF         AH1AF
                         115          (1) User Access Issues
                                          117
I1AFA        AH2AFA
                         117          (1) File Names
                                          117, 119, 127, 128, 129, 129, 131, 132, 132, 132, 133,
                                          133, 133, 134, 135, 136
I2AFA        AH2AFA
                         117          (1) User Access Issues
                                      (2) File Names
I1AFAA       AH3AFAA
                         117          (1) Symbolic Links
                                          118, 119
I2AFAA       AH3AFAA
                         117          (1) File Names
                                      (2) Symbolic Links
```

I1AFAAB	AH3AFAA	118	(1) User′s Point of View
I2AFAAB	AH3AFAA	118	(1) Symbolic Links (2) User′s Point of View
I1AFAAC	AH3AFAA	119	(1) ADSM Usage to Bypass the Symbolic Links
I2AFAAC	AH3AFAA	119	(1) Symbolic Links (2) ADSM Usage to Bypass the Symbolic Links
I1AFAB	AH3AFAB	119	(1) NFS Access 119, 119, 120, 121, 123
I2AFAB	AH3AFAB	119	(1) File Names (2) NFS Access
I1AFABA	AH3AFAB	119	(1) NFS Files Identity
I2AFABA	AH3AFAB	119	(1) NFS Access (2) NFS Files Identity
I1AFABB	AH3AFAB	119	(1) NFS Performance
I2AFABB	AH3AFAB	119	(1) NFS Access (2) NFS Performance
I1AFABC	AH3AFAB	120	(1) ADSM Configurations to Solve NFS Problems
I2AFABC	AH3AFAB	120	(1) NFS Access (2) ADSM Configurations to Solve NFS Problems
I1AFAB$	AH3AFAB	121	(1) Example of Use of Virtual Nodes
I2AFAB$	AH3AFAB	121	(1) NFS Access (2) Example of Use of Virtual Nodes
I1AFABD	AH3AFAB	123	(1) ADSM Client Interface
I2AFABD	AH3AFAB	123	(1) NFS Access (2) ADSM Client Interface
I1AG	AH1AG	125	(1) ADSM Availability
I1AGA	AH2AGA	127	(1) ADSM Availability Issues
I2AGA	AH2AGA	127	(1) File Names (2) ADSM Availability Issues
I1AGB	AH2AGB	128	(1) Mirror the Database and Log
I2AGB	AH2AGB	128	(1) File Names (2) Mirror the Database and Log
I1AGC	AH2AGC	129	(1) Dump/Load the ADSM Database
I2AGC	AH2AGC	129	(1) File Names (2) Dump/Load the ADSM Database
I1AGCA	AH2AGC	129	(1) How to Dump the Database
I2AGCA	AH2AGC	129	(1) File Names (2) How to Dump the Database
I1AGCB	AH2AGC	131	(1) How to Recover the Database
I2AGCB	AH2AGC	131	(1) File Names (2) How to Recover the Database
I1AGD	AH2AGD	132	(1) ADSM Auditing
I2AGD	AH2AGD	132	(1) File Names (2) ADSM Auditing
I1AGDA	AH2AGD	132	(1) Audit the ADSM Database
I2AGDA	AH2AGD	132	(1) File Names (2) Audit the ADSM Database
I1AGDB	AH2AGD	132	(1) Audit the Storage Volumes
I2AGDB	AH2AGD	132	(1) File Names (2) Audit the Storage Volumes
I1AGDC	AH2AGD	133	(1) Audit the Library

I2AGDC	AH2AGD	133	(1) File Names (2) Audit the Library
I1AGDD	AH2AGD	133	(1) Audit the License
I2AGDD	AH2AGD	133	(1) File Names (2) Audit the License
I1AGE	AH2AGE	133	(1) Export/Import Functions
I2AGE	AH2AGE	133	(1) File Names (2) Export/Import Functions
I1AGEA	AH2AGE	134	(1) Example One for Export/Import
I2AGEA	AH2AGE	134	(1) File Names (2) Example One for Export/Import
I1AGEB	AH2AGE	135	(1) Example Two for Export/Import
I2AGEB	AH2AGE	135	(1) File Names (2) Example Two for Export/Import
I1AGF	AH2AGF	136	(1) Standby ADSM Server 137, 137, 139, 140, 143
I2AGF	AH2AGF	136	(1) File Names (2) Standby ADSM Server
I1AGFA	AH2AGF	137	(1) ADSM/6000 Server Startup Operation
I2AGFA	AH2AGF	137	(1) Standby ADSM Server (2) ADSM/6000 Server Startup Operation
I1AGFB	AH3AGFB	137	(1) Takeover with Dump/Load Utility 138
I2AGFB	AH3AGFB	137	(1) Standby ADSM Server (2) Takeover with Dump/Load Utility
I1AGFAA	AH3AGFB	138	(1) Recovery Steps
I2AGFAA	AH3AGFB	138	(1) Takeover with Dump/Load Utility (2) Recovery Steps
I1AGFC	AH3AGFC	139	(1) Takeover with Export/Import Utility 140
I2AGFC	AH3AGFC	139	(1) Standby ADSM Server (2) Takeover with Export/Import Utility
I1AGFCA	AH3AGFC	140	(1) Recovery Steps
I2AGFCA	AH3AGFC	140	(1) Takeover with Export/Import Utility (2) Recovery Steps
I1AGFD	AH3AGFD	140	(1) Shared Disk Configuration
I2AGFD	AH3AGFD	140	(1) Standby ADSM Server (2) Shared Disk Configuration
I1AGFE	AH3AGFE	143	(1) Comparisons 143, 143, 143
I2AGFE	AH3AGFE	143	(1) Standby ADSM Server (2) Comparisons
I1AGFEA	AH3AGFE	143	(1) Load/Dump Utility
I2AGFEA	AH3AGFE	143	(1) Comparisons (2) Load/Dump Utility
I1AGFEB	AH3AGFE	143	(1) Export/Import Utility
I2AGFEB	AH3AGFE	143	(1) Comparisons (2) Export/Import Utility
I1AGFEC	AH3AGFE	143	(1) Shared Disk Configuration
I2AGFEC	AH3AGFE	143	(1) Comparisons (2) Shared Disk Configuration
I1AP1	AH1AP1	145	(1) ADSM Configuration Files 147, 155

I1AP2A	AH1AP2	147	(1) The adsmc Interface
I2AP2A	AH1AP2	147	(1) ADSM Configuration Files (2) The adsmc Interface
I1AP2C	AH1AP2	155	(1) SP2 Users Authorization
I2AP2C	AH1AP2	155	(1) ADSM Configuration Files (2) SP2 Users Authorization
I1AP3	AH1AP3	157	(1) Device Support

Tables

id	File	Page	References
ADSMST	AH3ADAE	71	1 70
ADSMSX	AH3ADBG	87	2 83, 86
IPPRMS1	AH3AECA	112	3
IPPRMS2	AH3AECA	112	4
HPSPRMS	AH3AECA	113	5

Processing Options

Runtime values:

Document fileid	GG244499 SCRIPT
Document type	USERDOC
Document style	IBMXAGD
Profile	EDFPRF40
Service Level	0022
SCRIPT/VS Release	4.0.0
Date	95.04.11
Time	13:10:48
Device	3820A
Number of Passes	3
Index	YES
SYSVAR G	INLINE
SYSVAR X	ALL

Formatting values used:

Annotation	NO
Cross reference listing	YES
Cross reference head prefix only	NO
Dialog	LABEL
Duplex	YES
DVCF conditions file	(none)
DVCF value 1	(none)
DVCF value 2	(none)
DVCF value 3	(none)
DVCF value 4	(none)
DVCF value 5	(none)
DVCF value 6	(none)
DVCF value 7	(none)
DVCF value 8	(none)
DVCF value 9	(none)
Explode	NO
Figure list on new page	YES
Figure/table number separation	YES
Folio-by-chapter	NO
Head 0 body text	Chapter
Head 1 body text	Section
Head 1 appendix text	Appendix
Hyphenation	NO
Justification	NO
Language	ENGL
Keyboard	395
Layout	OFF
Leader dots	YES
Master index	(none)
Partial TOC (maximum level)	4

```
Partial TOC (new page after) .................................. INLINE
Print example id′s ............................................ NO
Print cross reference page numbers ............................ YES
Process value ................................................. (none)
Punctuation move characters ................................... .,
Read cross-reference file ..................................... (none)
Running heading/footing rule .................................. NONE
Show index entries ............................................ NO
Table of Contents (maximum level) ............................. 3
Table list on new page ........................................ YES
Title page (draft) alignment .................................. RIGHT
Write cross-reference file .................................... (none)
```

Imbed Trace

```
Page 0                  4499SU
Page 0                    4499VARS
Page 0                  4499MAIN
Page 0                    ADVARS
Page 0                    4499FM
Page i                      4499EDNO
Page ii                     4499ABST
Page xiii                   4499SPEC
Page xiii                   4499TMKS
Page xiv                    4499PREF
Page xvi                    4499ACKS
Page xviii                AH1AA
Page 1                      AH2AAD
Page 3                        AH3AADB
Page 4                        AH3AADA
Page 5                        AH3AADC
Page 7                      AH2AAA
Page 9                        AH3AAAC
Page 13                       AH3AAAA
Page 17                       AH3AAAB
Page 18                     AH2AAB
Page 19                       AH3AABA
Page 21                       AH3AABB
Page 25                     AH2AAC
Page 30                   AH1AC
Page 31                     AH2AC#
Page 33                     AH2ACA
Page 35                       AH3ACAA
Page 37                       AH3ACAB
Page 40                     AH2ACB
Page 41                       AH3ACBA
Page 45                       AH3ACBB
Page 51                       AH3ACBC
Page 55                     AH2ACC
Page 57                       AH3ACCA
Page 58                       AH3ACCB
Page 61                   AH1AD
Page 63                     AH2ADA
Page 65                       AH3ADAA
Page 67                       AH3ADAB
Page 68                       AH3ADAC
Page 68                       AH3ADAD
Page 69                       AH3ADAE
Page 71                     AH2ADB
Page 74                       AH3ADBB
Page 76                       AH3ADBC
Page 78                       AH3ADBD
Page 79                       AH3ADBE
Page 80                       AH3ADBF
Page 83                       AH3ADBG
Page 87                   AH1AE
Page 89                     AH2AEB
Page 91                       AH3AEBA
Page 91                       AH3AEBC
Page 94                       AH3AEBD
Page 97                       AH3AEBE
Page 97                       AH3AEBF
Page 99                       AH3AEBG
Page 99                       AH3AEBH
Page 101                      AH3AEBI
Page 102                    AH2AED
Page 109                    AH2AEC
Page 111                      AH3AECA
Page 113                  AH1AF
Page 115                    AH2AFA
Page 117                      AH3AFAA
```

/XRL/17