Getting Started with ADSM NetWare Clients

Document Number GG24-4242-00

October 1994

International Technical Support Organization
San Jose Center

Take Note!

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

First Edition (October 1994)

This edition applies to Release 2 of ADSTAR Distributed Storage Manager (5648-020) and all subsequent releases and modifications unless otherwise indicated.

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. 471 Building 70B
5600 Cottle Road
San Jose, California 95193-0001

Abstract

This document describes the ADSTAR Distributed Storage Manager (ADSM) client for Novell NetWare. It provides information for the installation, customization, and use of the ADSM NetWare client. It covers the types of data supported by ADSM backup and archive functions, the connectivity options of client-to-server communications, requirements for running the ADSM client in the Novell NetWare environment, and backup and recovery scenarios.

This document is written for ADSM administrators, Novell NetWare specialists and administrators, LAN and networking specialists, storage administrators, and technical personnel interested in the ADSM client.

A limited knowledge of ADSM and Novell NetWare is required.

ST (123 pages)

Contents

Abstract . . . iii

Special Notices . . . xi

Preface . . . xiii
How This Document Is Organized . . . xiii

Related Publications . . . xv
International Technical Support Organization Publications . . . xvi

Acknowledgments . . . xvii

Chapter 1. Introduction to ADSM . . . 1
1.1 Structure . . . 1
1.1.1 Client/Server . . . 1
1.1.2 Storage Management . . . 2
1.1.3 Enterprise Support . . . 4
1.1.4 Communications Protocols . . . 6
1.2 Components . . . 7
1.2.1 Server . . . 7
1.2.2 Backup/Archive Client . . . 7
1.2.3 User Interfaces . . . 8
1.2.4 Storage Administration Function . . . 8
1.2.5 Administrative Client . . . 8
1.2.6 Security Function . . . 9
1.2.7 Application Client . . . 10
1.3 Storage Management Services . . . 10
1.3.1 Backup Processing . . . 10
1.3.2 Restore Processing . . . 11
1.3.3 Archive Processing . . . 11
1.3.4 Retrieve Processing . . . 12
1.3.5 General File Support . . . 12
1.3.6 Policy Management . . . 13
1.3.7 Function Execution . . . 18
1.3.8 Client Options File . . . 19
1.4 ADSM Support for NetWare . . . 21
1.4.1 Supported Environments . . . 21
1.4.2 Functions Not in NetWare ADSM Client Support . . . 21
1.4.3 Unique NetWare ADSM Client Capability . . . 22

Chapter 2. Introduction to NetWare . . . 23
2.1 NetWare Versions 3.11 and 3.12 . . . 23
2.1.1 Server Architecture . . . 23
2.1.2 Loadable Modules . . . 24
2.1.3 File System . . . 26
2.1.4 Security . . . 29
2.1.5 Users . . . 31
2.1.6 User Interface . . . 32
2.1.7 Connectivity . . . 33
2.1.8 Availability Features . . . 34
2.1.9 Storage Management Services . . . 35

2.2 NetWare Version 4 . . . 37
2.2.1 Directory Services . . . 37
2.2.2 File System . . . 38
2.2.3 Changes to SMS . . . 39
2.3 NetWare for SAA . . . 40
2.3.1 Packaging . . . 40
2.3.2 Run-time Versions . . . 41

Chapter 3. Implementing the ADSM NetWare Client . . . 43
3.1 Installing the ADSM Client . . . 43
3.1.1 Updating the Server AUTOEXEC.NCF File . . . 44
3.1.2 Updating the ADSM Options File . . . 45
3.2 IPX/SPX Connectivity . . . 48
3.2.1 System Requirements . . . 48
3.2.2 Configuration . . . 48
3.2.3 Problem Determination . . . 50
3.2.4 Performance Considerations . . . 50
3.3 TCP/IP Connectivity . . . 51
3.3.1 System Requirements . . . 51
3.3.2 Configuration . . . 52
3.3.3 Problem Determination . . . 53
3.3.4 Performance Considerations . . . 54
3.4 APPC Connectivity . . . 56
3.4.1 System Requirements . . . 58
3.4.2 Configuration . . . 60
3.4.3 Problem Determination . . . 69
3.4.4 Performance Considerations . . . 72
3.5 Starting the NetWare ADSM Client . . . 74

Chapter 4. Using the ADSM NetWare Client . . . 77
4.1 Backing Up and Restoring Data . . . 77
4.1.1 Incremental Backup . . . 77
4.1.2 Selective Backup . . . 78
4.1.3 Restoring Backup Data . . . 79
4.1.4 Backup and Restore of the NetWare 3.1x Bindery . . . 82
4.1.5 Backup and Restore of the NetWare 4.x Directory . . . 84
4.2 Archive and Retrieving Data . . . 85
4.2.1 Archiving . . . 85
4.2.2 Retrieving . . . 86
4.2.3 Deleting Archive Data . . . 86
4.3 Running Scheduled Operations . . . 87
4.3.1 Scheduling Related Client Options . . . 89

Chapter 5. Exploiting the ADSM NetWare Client . . . 91
5.1 Multiple ADSM Clients on a Single NetWare Server . . . 91
5.1.1 Multiple ADSM Clients with Different ADSM Servers . . . 92
5.2 Managing Multiple NetWare Servers with an ADSM Client . . . 94
5.2.1 Configuration and Use . . . 94
5.2.2 Operational Considerations . . . 98
5.2.3 Performance Considerations . . . 99
5.2.4 Summary . . . 100
5.3 NetWare Server Recovery with ADSM . . . 100
5.3.1 Using Multiple NetWare ADSM Clients . . . 101
5.3.2 Multiple ADSM Clients with Remote NetWare Servers . . . 103

Appendix A. Sample NetWare APPC Configuration Definitions 107
A.1 VTAM Start List 107
A.2 VTAM Application Major Node 107
A.3 VTAM Logmode Table 108
A.4 VTAM Workstation Definitions 108
A.4.1 VTAM 3745 Switched Major Node Definitions 109
A.4.2 VTAM Cross Domain Definitions 109
A.4.3 VTAM 3174 Local Node Definitions 109

Appendix B. Sample NetWare Server Configuration Files 111
B.1 IPX/SPX Connectivity 111
B.1.1 STARTUP.NCF 111
B.1.2 AUTOEXEC.NCF 111
B.2 TCP/IP Connectivity 111
B.2.1 STARTUP.NCF 112
B.2.2 AUTOEXEC.NCF 112
B.3 APPC Connectivity 112
B.3.1 STARTUP.NCF 112
B.3.2 AUTOEXEC.NCF 112

Appendix C. Sample NetWare PBTRACE 115

Appendix D. Sample NetWare CSSTATUS Trace 119

Index 123

Figures

1. NetWare As an ADSM Client . . . 2
2. ADSM MVS and VM Servers . . . 4
3. ADSM AIX/6000 Server . . . 5
4. ADSM OS/2 Server . . . 5
5. Administrative Client GUI . . . 9
6. ADSM Policy Management . . . 14
7. Policy Domains . . . 15
8. Policy Sets . . . 16
9. Copy Groups . . . 17
10. SERVER.EXE and the NLM Software Bus . . . 25
11. NetWare File System Overview . . . 27
12. NetWare Standard Directory Structure . . . 28
13. NetWare Trustee Rights Assignments . . . 30
14. NetWare User Interfaces . . . 32
15. NetWare Connectivity . . . 33
16. NetWare Server SMS Components . . . 36
17. NetWare Directory Services Tree Structure . . . 38
18. Changes to SMS Function with NetWare Version 4 . . . 39
19. NetWare Server Overview . . . 49
20. TCP/IP Sliding Window . . . 55
21. APPC Configuration Parameters . . . 57
22. APPC Options for ADSM NetWare Client and MVS Server . . . 59
23. NetWare ADSM Client with Multiple ADSM Directories . . . 93
24. Two NetWare Servers with a Single ADSM Client . . . 95
25. NetWare Overview . . . 101
26. Multiple ADSM Clients Managing Remote NetWare Servers . . . 104

Special Notices

This publication is intended to help ADSM administrators, Novell NetWare specialists and server administrators, LAN and networking specialists, storage administrators and other technical personnel to install, customize, and exploit the ADSM client. The information in this publication is not intended as the specification of any programming interfaces that are provided by ADSTAR Distributed Storage Management (5648-020). See the PUBLICATIONS section of the IBM Programming Announcement for ADSTAR Distributed Storage Management 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, 208 Harbor Drive, Stamford, CT 06904 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.

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, 208 Harbor Drive, Stamford, CT 06904 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

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, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ADSTAR	AIX
AIX/6000	IBM
NetView	OS/2
RACF	RISC System/6000
SAA	VTAM

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

DEC	Digital Equipment Corporation
HP-UX	Hewlett-Packard Company
IPX/SPX	Novell, Inc.
Macintosh	Apple Computer, Inc.
NetWare	Novell, Inc.
Novell	Novell, Inc.
SCO	The Santa Cruz Operation, Inc.
Solaris	Sun Microsystems, Inc.
SPARC	SPARC International, Inc.
Sun	Sun Microsystems, Inc.
Windows	Microsoft Corporation
ULTRIX	Digital Equipment Corporation
UNIX	X/Open Company, Ltd.

Preface

This document provides information for the installation, customization, and use of the ADSM client for Novell NetWare. It covers the types of data supported by ADSM backup and archive functions in the Novell NetWare environment, connectivity options of client/server communications, requirements for running the Novell NetWare ADSM client, and backup and recovery scenarios.

This document is intended for ADSM administrators, Novell NetWare specialists and administrators, LAN and networking specialists, storage administrators, and technical personnel interested in the ADSM client. A limited knowledge of ADSM and Novell NetWare is required.

How This Document Is Organized

The document is organized as follows:

- Chapter 1, “Introduction to ADSM” provides an overview of ADSM.
- Chapter 2, “Introduction to NetWare” provides an overview of the NetWare operating environment.
- Chapter 3, “Implementing the ADSM NetWare Client” provides information about planning for, connecting to, and implementing the ADSM backup and archive client.
- Chapter 4, “Using the ADSM NetWare Client” explains how to use the ADSM backup and archive client.
- Chapter 5, “Exploiting the ADSM NetWare Client” describes how to exploit the ADSM client, including how to use ADSM in various backup and recovery scenarios.
- Appendix A, “Sample NetWare APPC Configuration Definitions” contains sample parameters for APPC connection between the ADSM NetWare client and server.
- Appendix B, “Sample NetWare Server Configuration Files” contains samples of various ADSM NetWare client startup files.
- Appendix C, “Sample NetWare PBTRACE” contains a sample PBTRACE for an APPC session.
- Appendix D, “Sample NetWare CSSTATUS Trace” contains a sample CSSTATUS trace for an APPC session.

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: User's Guide and Reference for Novell NetWare*, SH35-0124
- *ADSTAR Distributed Storage Manager: General Information*, GH35-0114
- *ADSTAR Distributed Storage Manager: Administrator's Guide*, SH35-0117
- *ADSTAR Distributed Storage Manager/2: Administrator's Guide*, SH26-4003
- *ADSTAR Distributed Storage Manager/6000: Administrator's Guide*, SH26-4005
- *ADSTAR Distributed Storage Manager: Administrator's Reference*, SH35-0130
- *ADSTAR Distributed Storage Manager/2: Administrator's Reference*, SH26-4004
- *ADSTAR Distributed Storage Manager/6000: Administrator's Reference*, SH26-4006
- *ADSTAR Distributed Storage Manager/2: Installing the Server and Administrative Client*, SH26-4014
- *ADSTAR Distributed Storage Manager/6000: Installing the Server and Administrative Client*, SH26-4013
- *ADSTAR Distributed Storage Manager: Administration Messages*, SH35-0129
- *ADSTAR Distributed Storage Manager: Performance Tuning Guide Release 2*, SH26-4034
- *ADSTAR Distributed Storage Manager Presentation Guide*, GG24-4146
- *ADSTAR Distributed Storage Manager Storage Management Services: Implementation Examples*, GG24-4034
- *ADSTAR Distributed Storage Manager Advanced Implementation Experiences*, GG24-4221
- *Getting Started with ADSM/2*, GG24-4321
- *Getting Started with ADSM/6000*, GG24-4421
- *The Host As a Data Server Using LANRES and Novell NetWare*, GG24-4069
- *NetWare in the TCP/IP and UNIX Environment*, GG24-3893
- *APPC/MVS Performance Observations for MVS/ESA SP Version 4.3*, GG66-3206
- *Multi-platform APPC Configuration Guide*, SV40-0089
- *Novell NetWare Version 3.11 Installation*, 183-000298
- *Novell NetWare Version 3.11 System Administration*, 183-000296
- *Novell NetWare Version 3.11 Utilities Reference*, 183-000293
- *Novell NetWare Version 3.11 System Messages*, 183-000295
- *Novell NetWare Version 3.11 TCP/IP Transport Supervisor's Guide*, 100-000945
- *Novell NetWare Version 3.11 Server Backup*, 100-000951

- *Novell NetWare Version 3.11 Concepts*, 183-000294
- *Novell NetWare Version 3.11 Requester for OS/2*, 100-001157
- *Novell NetWare for SAA 1.3 Rev. B Administration Guide*, 100-001166
- *Novell NetWare Software Developer's Kit, APPC Technical Reference*, 100-001101
- *Novell NetWare Version 4.0 Installation and Upgrade*, 74G1412
- *Novell NetWare Version 4.0 Getting Started with NetWare 4.0*, 74G1413

International Technical Support Organization Publications

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.

The books that may be of particular interest to someone installing ADSM include:

ADSTAR Distributed Storage Manager Presentation Guide, GG24-4146

ADSTAR Distributed Storage Manager Storage Management Services: Implementation Examples, GG24-4034

ADSTAR Distributed Storage Manager Advanced Implementation Experiences, GG24-4221

Getting Started with ADSM/2, GG24-4321

Getting Started with ADSM/6000, GG24-4421

Using ADSM to Back Up Databases, GG24-4335

The Host As a Data Server Using LANRES and Novell NetWare, GG24-4069

NetWare in the TCP/IP and UNIX Environment, GG24-3893

Acknowledgments

This publication is the result of two residency projects conducted at the International Technical Support Organization, San Jose Center. The first project established the ADSM Novell NetWare environments. The second residency produced this book.

The first project was designed and managed by:

Sally J. Montera, Storage Products Assignee, ITSO - San Jose Center.

The second residency was managed by:

Dale Freeman, Storage Products Assignee, ITSO - San Jose Center.

The author of this document is:

Tim Mortimer, IBM UK

The book is based on implementation work by:

Mark Blunden, IBM Australia
Walter Majonica, IBM Germany
Carolina von Hinrichs, IBM Germany.

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

Pete Alcantara, IBM U.S. Marketing, Area 11
Cybelle Beaulieu, IBM Storage Systems Division
Maggie Cutler, Technical Editor
Len Ling, IBM Storage Systems Division
Tom Miller, Novell, Inc.
Ping Mong, IBM Storage Systems Division
Frank Ramke, IBM Storage Systems Division
Lenora Wang, IBM U.S. Marketing, Storage Services

International Technical Support Organization, San Jose Center.

October 1994

Chapter 1. Introduction to ADSM

ADSTAR* Distributed Storage Manager (ADSM) is an IBM* product that provides backup, recovery and space management services for user and system data on a wide variety of computing platforms.

The objective of this chapter is to introduce ADSM to users and administrators of NetWare** and to telecommunications and local area network (LAN) support specialists who may be involved in the implementation of ADSM in a NetWare environment. If you are already familiar with ADSM′s concepts and facilities, you may want to go directly to Chapter 2.

This chapter is not intended to be a comprehensive discussion of ADSM, but it provides sufficient background information to enable you to understand the concepts and facilities of NetWare ADSM clients. Emphasis is on those ADSM features that are relevant to the NetWare system.

The focus of the entire book is on the NetWare ADSM client. It is assumed that an ADSM server that supports NetWare is available and that an ADSM NetWare supported communications protocol product is installed on the ADSM server and client.

1.1 Structure

ADSM provides storage management services for data in a client/server environment. Many client and server platforms and communications protocols are supported by ADSM. The details of this support are described below.

1.1.1 Client/Server

ADSM is a client/server product. Part of ADSM runs as the client, typically on a programmable workstation (PWS), and part of ADSM runs as the server, usually in a larger machine. A user of ADSM has data on a client machine that uses the storage management services that ADSM provides on a server machine.

In a NetWare system the terms *client* and *server* are also used. A NetWare environment includes a requester, typically a PWS, connected to a NetWare server machine. The requester can also be called a NetWare client. Throughout this book the NetWare client is called the requester. The NetWare server provides data storage, printer, and application serving support to the requester. Thus, in an ADSM context, a NetWare server can also be an ADSM client as is shown in Figure 1 on page 2.

In Figure 1 on page 2, the machine at the top is an ADSM server, perhaps running MVS. The two machines on the left are NetWare servers, although the upper left machine is also an ADSM client. The two machines on the right are NetWare requesters.

Users at either the DOS or OS/2* requester could be storing data and using programs on either of the two NetWare servers. The users also could use ADSM

services to back up data from the ADSM client to the ADSM server. Data on the lower left NetWare server can also be backed up by the ADSM client running on the machine shown in the upper left.

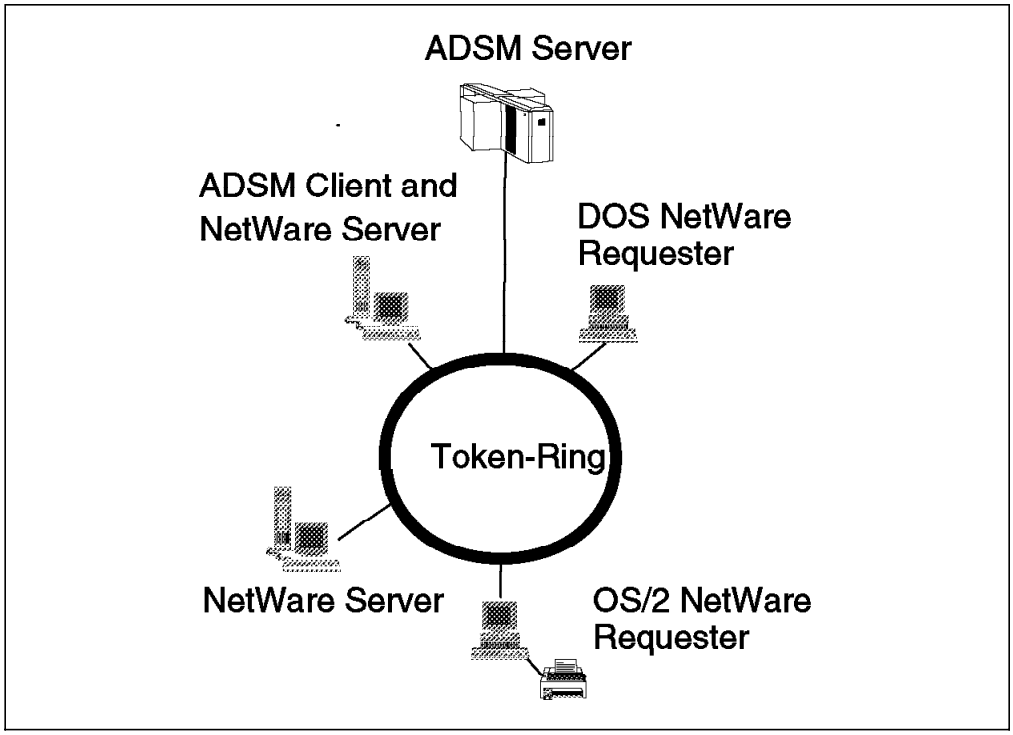


Figure 1. NetWare As an ADSM Client

In this book we use the following:

- Requester - always means the NetWare requester.
- Client - always means the ADSM client, which in the ADSM NetWare environment is also a NetWare server.
- Server - is qualified as either a NetWare server or an ADSM server.

The term *node* is also used within ADSM to identify an ADSM client. A *node name* is the name given to a particular ADSM client or node. It is used like a user ID when the system connects to the ADSM server.

Another term that needs to be clarified when using ADSM and NetWare is *directory*. Directories are commonly used in workstation environments, including ADSM with NetWare, to represent a group of files. With NetWare version 4 the directory has an additional meaning, namely, the system information used to locate any NetWare version 4 resource.

1.1.2 Storage Management

The major storage management functions that ADSM provides are:

- Backup and restore
- Archive and retrieve.

These functions can also be confusing. By *backup*, ADSM means creating a version of a file to be used for recovery. This backup version is stored separately, on the ADSM server, from the master copy of the data. Potentially, you can make several backup versions of the data, each version at a different point in time. These versions are closely tied together and to the original file as a group of backups.

If the master file is invalid or lost, *restore* is the process of using a backup version of the data to re-create the master copy. The most current version of the data would normally be used, but you can restore from any of the backup versions that exist.

The number of backup versions for a file is normally controlled. Old backup versions may be automatically deleted as new versions are created. You may also delete them after a certain period of time.

By *archive*, ADSM means creating a copy of a file as a separate file. Usually you would use this function to create an additional copy of the data to be saved for historical reasons. This type of data is called vital records, that is, data that must be kept by law or for other business reasons.

As this copy of data is created on the server, you can delete the original copy. Thus, you can use archive to make additional space available on the ADSM client. Archive should not be thought of as a complete space management function because automatic recall is not available.

You can access archived data by *retrieve* to return it to the ADSM client if the data is needed at some future time. To help determine which archived data is required, you can add a description to the data at the time the file is archived. Later, when you want the data again, the description can be used for a search to determine which file to retrieve.

Thus, the difference between backup and archive is that backup creates and controls multiple backup versions that are directly attached to the original file, whereas archive creates an additional file that is normally kept for vital records.

ADSM provides functions that automate backup and archive based on predefined policies. You can build these policies to tailor services for ADSM users and allow the selection of data to include or exclude from either backup or archive processing.

Data recovery includes recovery of:

- Incorrectly modified data
- Prematurely deleted data
- Data lost because of disk, media, or other hardware failures
- Data lost because of site damage or disaster.

The backup and restore facilities of ADSM provide support for all such recoveries. Be aware that other recovery tools and processes may be required in certain situations. For example, before ADSM can recover your data, it and your client's operating system must be up and running on your machine. ADSM cannot be used to recover itself, so other tools must be used.

ADSM provides additional function to support the recovery associated with site damage or disaster. You may want to remove selected data, using *export/import* services, from the ADSM server's site and store it at another location so it will not be damaged during a disaster. This function is only available for data on the ADSM server.

An additional component of ADSM is the storage administrator. The storage administrator is one or more persons who determine how data should be managed within ADSM through tailored policies for backup and restore of data between the ADSM client and ADSM server.

1.1.3 Enterprise Support

One of the key advantages of ADSM is that it can manage storage in your entire enterprise. ADSM supports a wide variety of client/server platforms as shown in Figure 2, Figure 3 on page 5, and Figure 4 on page 5.

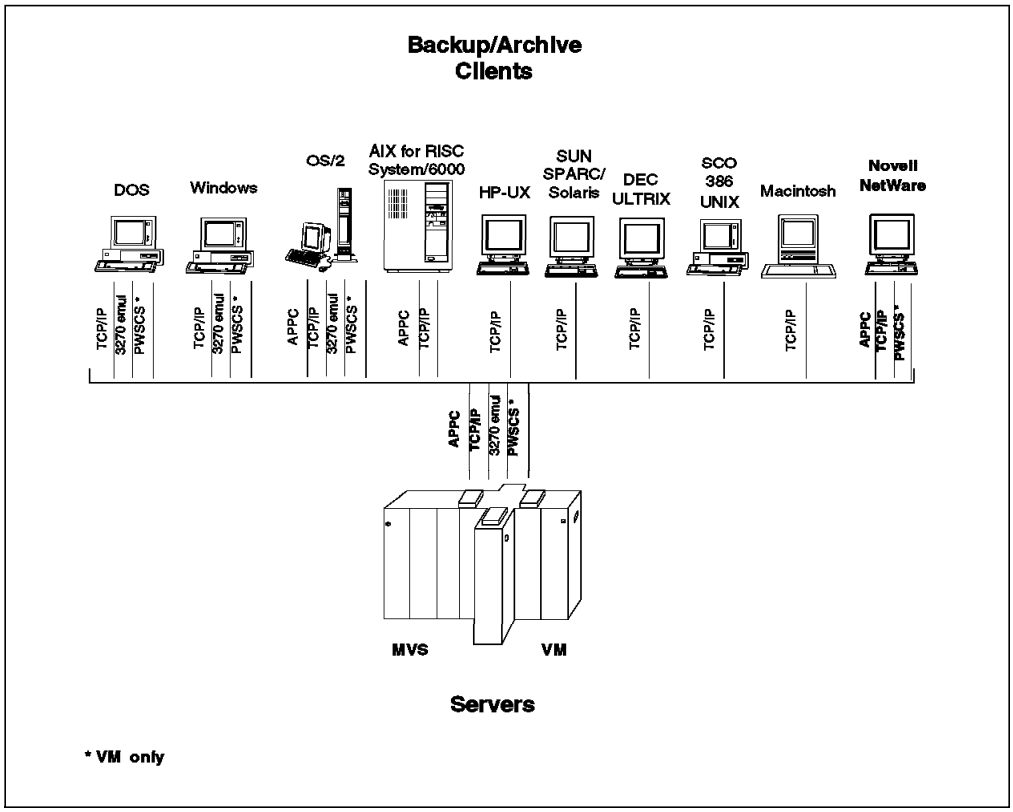


Figure 2. ADSM MVS and VM Servers

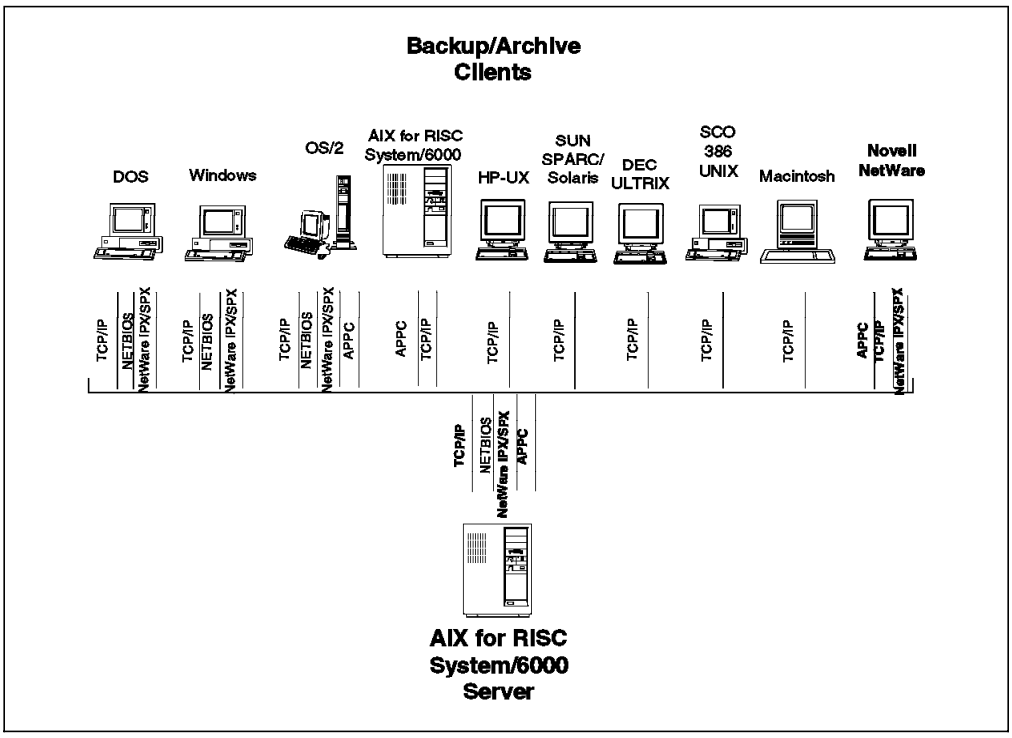


Figure 3. ADSM AIX/6000 Server

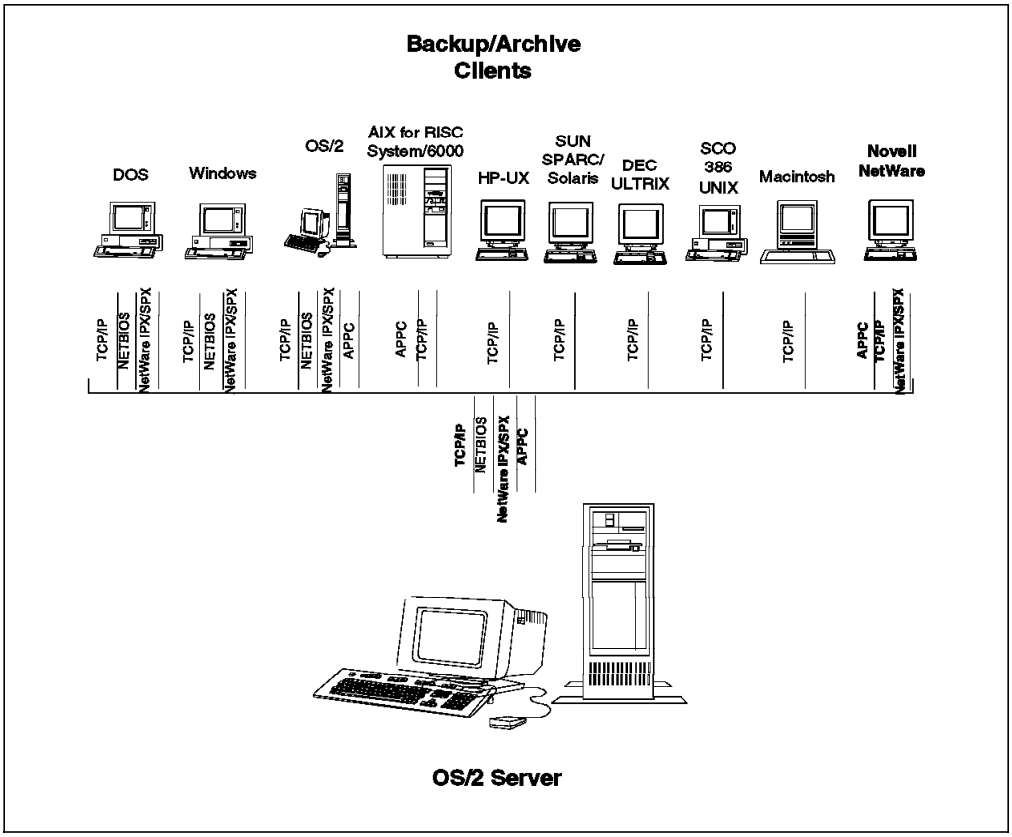


Figure 4. ADSM OS/2 Server

NetWare is currently supported as an ADSM client to four ADSM servers using a wide variety of communications protocols. The NetWare environments for ADSM include:

- MVS server support using TCP/IP and APPC communications protocols.
- VM server support using TCP/IP, APPC, and PWSCS communications protocols.
- AIX*/6000 server support using TCP/IP, APPC, and IPX/SPX** communications protocols.
- OS/2 server support using TCP/IP, APPC, and IPX/SPX communications protocols.

With each future release of ADSM, additional servers and clients will be added, so that you can have a single storage management strategy for all of your data.

1.1.4 Communications Protocols

As seen in the above discussion, a wide variety of communications protocols can be used to connect the ADSM client with an ADSM server. Up to four communications protocols are available with NetWare. The protocol you select depends on the type of ADSM server that is being used.

TCP/IP is the communications protocol commonly used in most AIX and UNIX** environments. It is an open system, standardized capability for sending information between common or dissimilar systems in a heterogeneous environment. All ADSM servers support NetWare ADSM clients using TCP/IP.

APPC is part of IBM's Systems Network Architecture (SNA). It allows two partner programs to communicate with each other as peers. APPC is available between all ADSM servers and NetWare ADSM clients.

IPX/SPX is Novell's** own proprietary communications protocol. It is used for communications between the NetWare requester and server. It is also used for communications between NetWare functional components. ADSM provides support to connect a NetWare ADSM client to an OS/2 ADSM server using IPX/SPX.

PWSCS is a VM communications protocol. Although it is supported between a NetWare ADSM client and VM as an ADSM server, it is not discussed in this book.

In the case where an ADSM server can use one of several protocols, the choice will almost always be based on the enterprise's overall strategy and the in-place communications products, particularly on the server. TCP/IP and APPC appear to be the two most widely used protocols with NetWare when using ADSM.

Although ADSM is a storage management product, the communications protocol component is one of the most important areas for successful implementation of ADSM. It is mandatory to coordinate the communications protocol parameters for the ADSM client and the ADSM server as well as have them fit with other uses, if any, of the communications protocol.

1.2 Components

In this section we introduce specific components of ADSM. Not all components are presented because this discussion is meant to be an introduction to the implementation of the ADSM NetWare client. In particular, detailed descriptions of the components of an ADSM server are omitted. They can be found in the ADSM administrator's guide for the particular server platform of interest and are listed in the related publications section of this book.

1.2.1 Server

The main ADSM functions are provided by the ADSM server. Currently ADSM has servers for MVS, VM, OS/2, and AIX systems. Customer requirements indicate that additional servers will be added over time.

Each server is implemented in a client-independent manner; that is, the server provides a standard set of functions independent of the type of ADSM client that is being supported. The ADSM server uses different device types and communications protocols appropriate to its base platform. For example, the ADSM MVS server uses 3420, 3480, or 3490 tape drives, whereas the ADSM OS/2 server uses a variety of SCSI-attached tape products including 8200, 8500, and 8505.

The main role of the ADSM server is to store the backup or archive data from the ADSM clients it supports. It also has a database of information to keep track of the backup and archive data that it manages.

The ADSM server may use a variety of device types to store the backup and archive data received from its ADSM clients. The device types can be arranged in an ADSM-managed storage hierarchy for a cost-efficient way of providing the best support for ADSM users. For example, you can have three storage pools, one made up of tape devices, one made up of DASD devices, and one made up of optical storage devices, with automated movement of the backup and archive data from one storage pool to another. This allows less used data to reside on the less expensive storage medium.

ADSM servers are stand-alone servers. An enterprise may have many ADSM servers, on either different platforms or the same platform, but each ADSM server is independent of all other ADSM servers. There is no hierarchical or distributed ADSM server support at this time.

1.2.2 Backup/Archive Client

ADSM has several client programs. These programs normally run at the user's workstation or on a LAN server machine to provide ADSM services. The most important of these is the backup/archive client.

The ADSM *backup/archive client* is the program that allows users to register their workstations with an ADSM server as a client node. Then, users can maintain backup versions of workstation (ADSM client) files, which can be restored if the original files are lost or damaged. Users can also archive files

that they no longer need on their workstation and retrieve the archived files only when necessary.

1.2.3 User Interfaces

You have have two methods of interacting with the system from your ADSM client workstation: the command line interface (CLI) and the graphical user interface (GUI). The CLI consists of a set of specific commands with a syntax similar to other commands that you issue from the ADSM client platform.

The GUI enables you to interact with the system through windows and action bars. Both interfaces are available for most ADSM clients, but currently only the CLI is available for the NetWare ADSM client.

1.2.4 Storage Administration Function

ADSM has two type of users. The first is the typical user, who has data on the ADSM client system and wants ADSM server support for that data. In the NetWare environment this user is the person who works at a NetWare requester and uses the NetWare client as a data, print, or application server.

The second type of ADSM user is an ADSM administrator. An ADSM implementation can have one or more administrators, each with his or her own administrative functions and ADSM capabilities. Administrator functions include:

- Registering new users and their ADSM client systems
- Managing security for the ADSM environment
- Managing the storage management policies that provide tailored backup and archive support for user's data on the ADSM client systems
- Managing the scheduling of automated backup and archive
- Determining the hierarchy of administrators (if necessary) and their specific authorities
- Initiating the disaster recovery export/import function for data stored on the ADSM server.

Several levels of authority can be given to an administrator. There must be at least one high-level administrator who has authority over the entire system.

1.2.5 Administrative Client

The administrative functions can be invoked from the ADSM server's system consoles. A more user-friendly interface is another ADSM client, the *administrative client*. This program provides functions that enable a storage administrator to:

- Control and monitor ADSM server activities
- Define storage management policies for ADSM client workstation files
- Set up schedules to provide backup and archive services at regular intervals.

The administrative client can be run from an ADSM client or ADSM server workstation. Figure 5 on page 9 shows the GUI for the administrative client functions. A command line interface for the administrative client is also available.

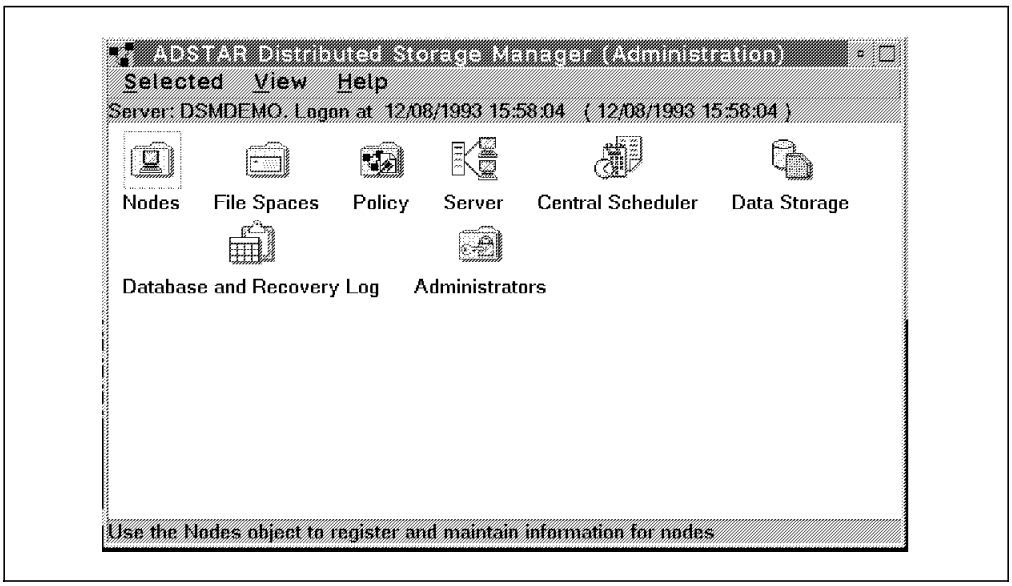


Figure 5. Administrative Client GUI

At this time, the administrative client is not part of the ADSM Netware support. Therefore the administrator functions must be done through the server. When MVS is the ADSM server for NetWare, the administrative client can be accessed remotely from a NetWare requester workstation through TSO.

1.2.6 Security Function

Before an ADSM client can begin requesting backup or archive services the client must be registered with the ADSM server. ADSM clients are referred to as nodes. Clients, or nodes, can be added to ADSM through either closed or open registration.

With *closed* registration, an administrator must register each ADSM client workstation in advance. With *open* registration, ADSM clients can register their own workstations with the ADSM server. When a user tries to access the ADSM server from an unregistered client node, ADSM prompts the user for a password and contact information to dynamically register the client.

ADSM provides client/server authentication to validate that administrative or backup/archive ADSM clients are communicating with an authorized ADSM server and that the server is communicating with registered clients. You could call this “mutually suspicious” validation because the ADSM client suspects that the server is not valid, and the ADSM server suspects that the clients are not valid.

ADSM can ensure authentication by requiring a password from each ADSM client. An optional process for some of the ADSM client platforms allows you to set up the system such that users are not required to enter a password every time authentication occurs.

ADSM does not currently use platform-specific authentication and authorization procedures because it was designed to provide a common interface across multiple platforms. Thus, for example, ADSM has no interaction with RACF* when MVS is used as the ADSM server platform.

1.2.7 Application Client

The *application client* program allows other IBM and non-IBM products to use the storage management services of ADSM with a standard interface. The application client runs on the ADSM client workstation and uses an application programming interface (API) to back up, archive, restore, or retrieve objects from an ADSM server.

1.3 Storage Management Services

This section discusses the storage management services provided to the ADSM client. Additional details on the backup/archive client and the storage administration policies for managing ADSM client files are provided.

1.3.1 Backup Processing

The main function of ADSM is backup/restore. You can back up all of your files (full), select specific files you want to back up (selective), or only back up the files that have changed since your last backup (incremental).

The backup process creates a copy of a client file, such as \MYFILE.DATA, on the ADSM server. The backup process also backs up the directory in which the file resides. You can request a backup by command, or ADSM can automatically create a backup. The administrator can establish a schedule where backup is done automatically.

All backups of a particular file are related and are called *versions*. The most current version is the *active* backup, and the other versions are *inactive* backups.

1.3.1.1 Selective Backup

You can specifically include or exclude certain files from being backed up. For example, you might not want everyone to back up their local copies of the NetWare operating system. For a selective backup, use INCLUDE and EXCLUDE statements, which are processed in a bottom-up order. Thus, if you have several INCLUDE and EXCLUDE statements, their order is important in determining which files are selected.

Selective backup specifies which files you wants to back up. A selective backup can consist of a single file, or you can select a directory or subdirectory tree to back up. Wildcards, such as “*,” are allowed in the specification, so there is great flexibility in file selection.

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

1.3.1.2 Incremental Backup

Incremental backup sends the files that have changed since the last backup to the server. The first time an incremental is done, all files are sent to the ADSM server. Incremental backup is the most common backup approach, particularly when it is done on an automatic schedule that the storage administrator has predefined.

Incremental backup is also controlled by the use of INCLUDE and EXCLUDE statements to filter in or out specific files that are considered for incremental processing. To be backed up the files need to be included and have changed since the last backup. An amount of time between backups can also be specified to prevent backups from occurring too often.

1.3.2 Restore Processing

Restore is the process of copying a backup version from the ADSM server to the ADSM client. This process is system assisted, which means that the system performs the restore at your request without having to call the ADSM administrator.

When you need to restore a file, you can either restore the latest backup copy or use an older version of the file. If you are restoring a file, ADSM provides help to prevent you from overlaying your current copy of the file.

Protection is built in to ADSM to prevent you from overwriting a file on your workstation during the restore process. A collision occurs when you restore a file and the original file still exists on the workstation. You can ask ADSM to prompt you every time a collision occurs so that you can respond on a file-by-file basis, or you can instruct ADSM never to overwrite the original files or always to overwrite the original files.

You can also protect yourself from overwriting the original files on your workstation by restoring the files to an alternative path. This would place the file into a different directory but maintain the same file name.

1.3.3 Archive Processing

ADSM has a separate archive function that lets you store files that you need to retain for long-term storage. Because the archived copy is expected to be retained for a long time, automatic deletion is not provided.

The archive process creates a copy of an ADSM client file, such as \MYFILE.DATA, on the ADSM server. Like backup, archived files are managed on the basis of policies, and archive can be initiated by user command or by ADSM automatic scheduling.

A key difference between backing up a file and archiving a file is that you can erase the original file after archiving it with no impact on the arhived copies. Archive can be used in this way to reduce the used disk space on your workstation.

Because the archive function does not use versioning, it can be more difficult for you to retrieve the desired copy of a file that has multiple archives. You can save a description of an archived file so that it will be easy to retrieve the file if multiple archived copies exist with the same file name.

1.3.4 Retrieve Processing

You use retrieve to obtain copies of files that you have archived on the ADSM server. You can retrieve a single file, a group of files, or all files in a directory or subdirectory. When you retrieve a file, ADSM provides support to prevent you from overwriting an existing copy on the ADSM client. The retrieved file remains intact on the ADSM server.

If a description was stored with the file on the ADSM server, it can be used to select the copy to retrieve. Selection using the description can include wildcards. The date when the file was archived can also be used to help determine which data to retrieve.

1.3.5 General File Support

In general, ADSM supports all the file types that are possible on each ADSM client. DOS, OS/2 Macintosh** and Windows** machines can be NetWare requesters. Thus, ADSM supports DOS, OS/2, NFS, FTAM, and Macintosh files while using NetWare as the ADSM client.

Although NetWare supports its requester′s file types, when the data is stored on the NetWare server, it is stored in a proprietary NetWare format. Thus, the files stored on the NetWare ADSM client are in NetWare file format even though the user sees them as DOS OS/2, Macintosh, or Windows files on the NetWare requester.

1.3.5.1 Data Compression

Data compression is an optional feature that you can use. The data is compressed on the ADSM client workstation, thus reducing the network traffic and the storage required on the ADSM server to hold the file. The ADSM compression algorithm on average compresses the file approximately 50%.

Compression reduces the amount of ADSM server data storage space and data that must be sent across the network between the ADSM client and server. The disadvantage of compression is that it requires additional processor time on the ADSM client workstation, and thus backup runs slower on the ADSM client.

1.3.5.2 Cross-Platform Restore and Retrieve

One powerful feature of ADSM is the ability to back up and restore or archive and retrieve from different client platforms. For example, you can back up your files from your DOS workstation and then restore them on an OS/2 workstation. This is very useful, for example, when all of your users migrate to a new platform or work out of multiple locations that have different client platforms.

The cross-platform restore and retrieve functions are not currently supported for data on the NetWare ADSM client. Cross-platform restore and retrieve works among DOS, Windows, and OS/2 clients, or among all UNIX clients (AIX for RISC System/6000*, Sun** SPARC** Solaris**, HP-UX**, DEC** ULTRIX**, and SCO** 386). It is not supported between UNIX and non-UNIX clients.

1.3.5.3 Cross-User Restore and Retrieve

Another powerful feature of ADSM is the ability to have other users restore or retrieve the files you backed up. You can authorize other users access to the files you have backed up or archived on a file-by-file basis. This capability is fully available for the NetWare user.

You use ADSM commands to grant other users access to your files. You establish access for cross-user restore and retrieve through these commands.

1.3.6 Policy Management

A key power of ADSM comes from its ability to use storage management policies to invoke automatic services. Both backup and archive are examples of ADSM client services that you can control by predefined policies. In each case, you, as the administrator, have can adjust the process by command parameters.

The granularity of control that you have is down to the file level. You can choose how granular you want your policies to be. You can establish an overall system policy, policies by department or organizational structure, or policies by user or file name. Policy management makes ADSM a true system-managed storage implementation.

The backup and archive functions can also be run on a predefined schedule. This feature, along with policy management, provides the power of ADSM. The ADSM administrator establishes the default policies and schedules for ADSM clients.

The elements of policy management are:

- Policy domains
- Policy sets
- Management classes
- Copy groups.

Figure 6 on page 14 shows the general relationship of these elements. Each is discussed in the sections that follow.

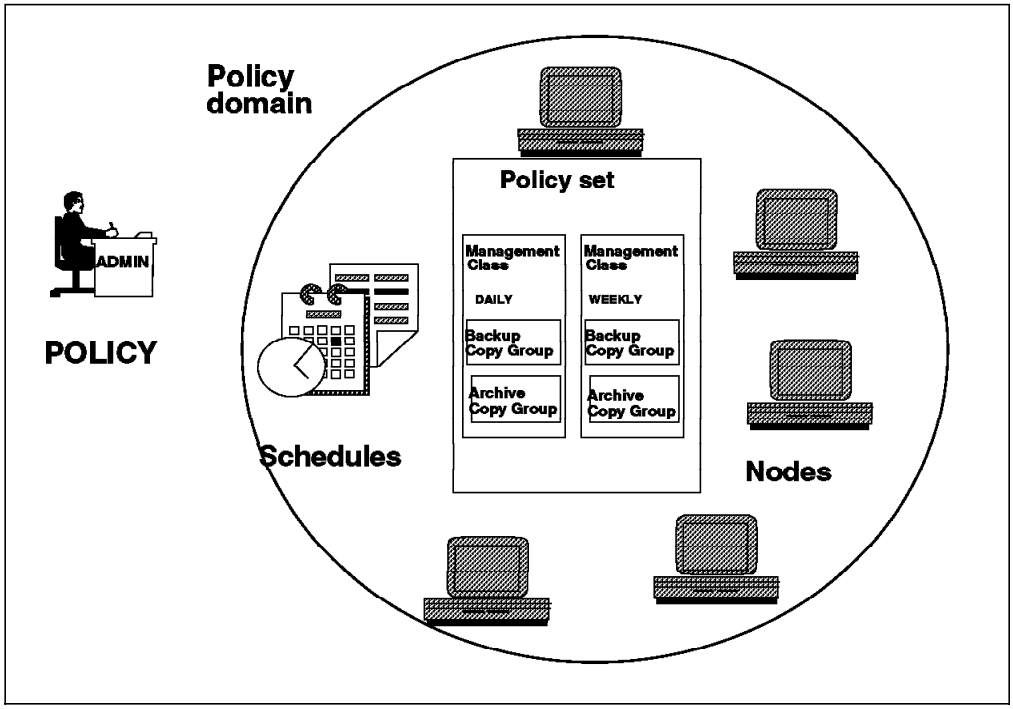


Figure 6. ADSM Policy Management

Policy management is both an ADSM client and server function. On the ADSM server the administrator is responsible for creating default policies. The ADSM client, however, can overwrite the default management class or select another management class in the client's policy domain. Clients can only select and use defined and activated policies; a client cannot define its own policies.

1.3.6.1 Policy Domain

A *policy domain* is a group of ADSM clients who are working according to the same set of policy needs. All or a group of NetWare ADSM clients might be associated with one domain, while AIX ADSM clients might be associated with another domain. Thus, policy domain provides a logical way of managing backup and archive policies for a group of client nodes.

Policy domains can be used to provide standard storage management policies to most users by grouping together ADSM clients that have similar storage management requirements. They can be used to limit the number of clients that need to be managed by a single policy administrator or restrict the number of management classes to which users have access.

A policy domain contains one or more policy sets. Each domain contains the following information:

- The domain's name
- The associated policy set(s)
- Grace period backup and archive retention periods.

  Grace period backup and archive retention periods act as a safety net to ensure that backed up and archived data cannot be accidentally deleted if it loses its copy group.

Figure 7 on page 15 shows three policy domains, which might reflect different departments in your organization or users with different data requirements. The first domain is used by a group of NetWare users and workstations. The second and third domains support different sets of users and workstation platforms.

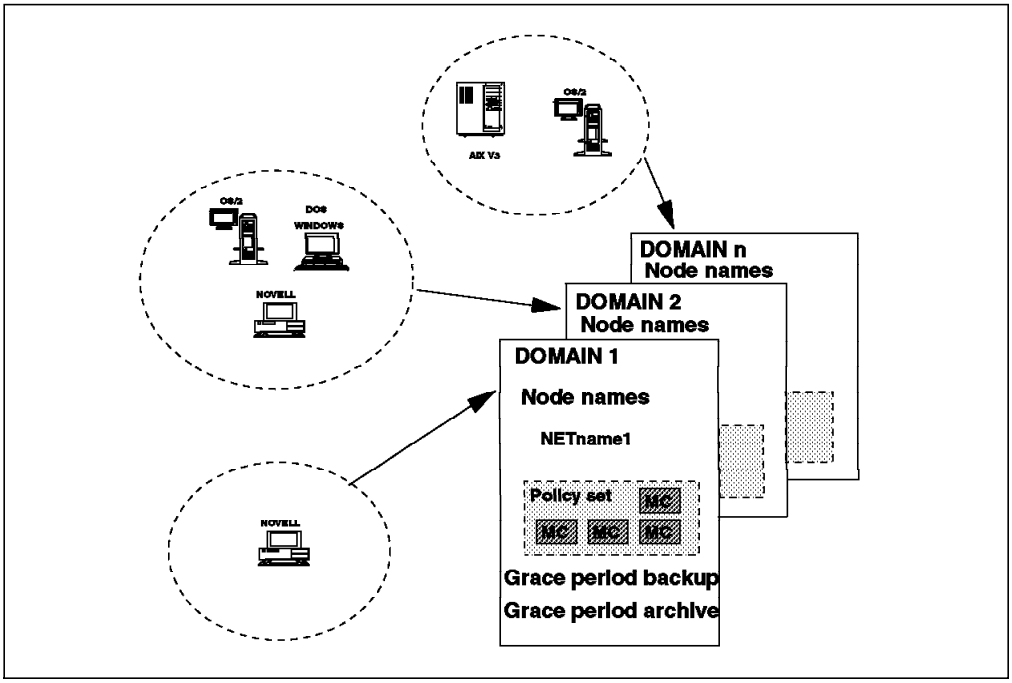


Figure 7. Policy Domains

1.3.6.2 Policy Set

A policy domain can have more than one *policy set*, but only one policy set can be activated at any point in time. In Figure 8 on page 16 you can see the ACTIVE policy set and both a NEW and OLD policy set. The OLD might be used for fall back, while the NEW policy set might still be in development for future requirements.

Each policy set contains a default management class and can contain any number of additional management classes. A management class typically contains a backup copy group and an archive copy group. Both management classes and copy groups will be discussed in more detail.

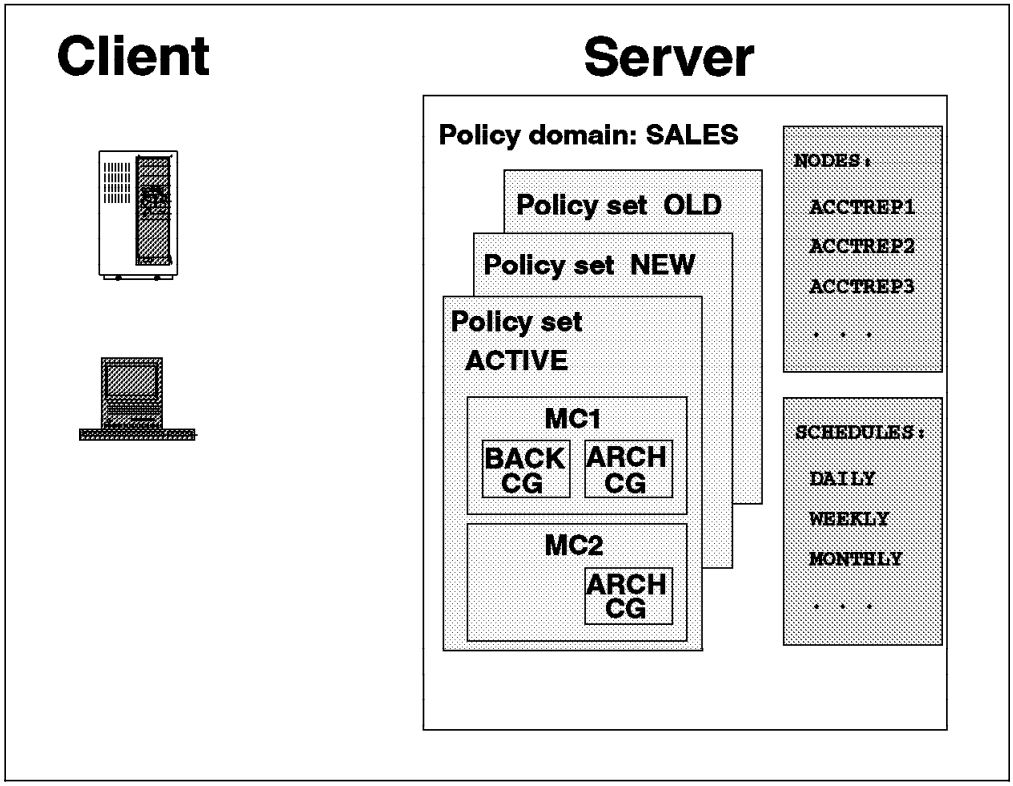


Figure 8. Policy Sets

1.3.6.3 Management Class

Policy sets contain one or more *management classes*. Management classes contain a backup copy group and/or an archive copy group or no copy group. The default management class must have a backup copy group. You can think of management classes as a Service Level Agreement you have with your clients on how their backup and archive data will be handled.

You can bind and thus associate a management class to a file when it is backed up or archived. You can rebind a file with a new management class. You can use the default management class or explicitly select a management class that is within the policy sets to which you have access. You select a management class other than the default by use of the INCLUDE statement.

1.3.6.4 Copy Group

You specify in *copy groups* the parameters that control the generation and expiration of backup and archive data. There is a separate copy group for backup and one for archive. Figure 9 on page 17 shows a backup and an archive copy group. The values shown are the defaults that are provided as part of ADSM.

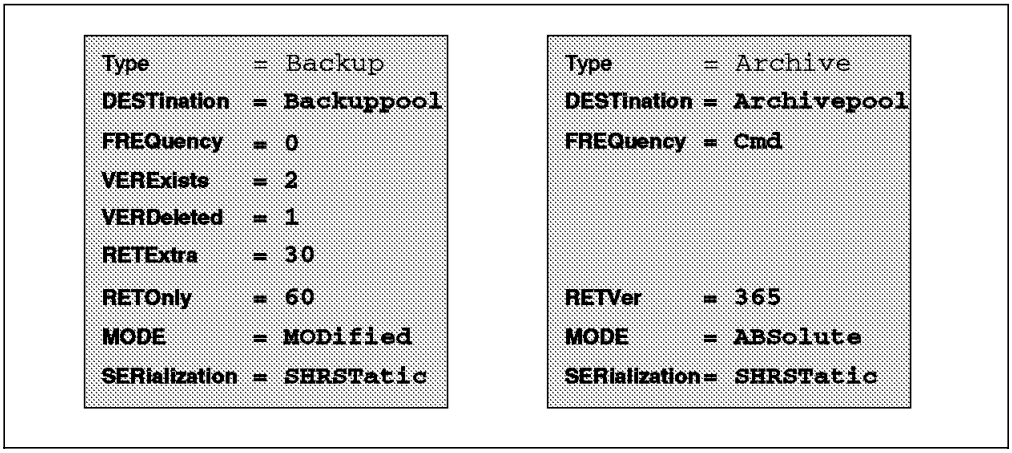


Figure 9. Copy Groups

Both backup and archive copy groups have similar parameters, except that the version parameters apply only to backup files. The parameters are:

DESTINATION Specifies the name of the storage pool where the ADSM server initially stores the backed up or archived files. The default destination for a backup copy group is BACKUPPOOL. The default destination for an archive copy group is ARCHIVEPOOL.

FREQUENCY Specifies, for a backup file, the minimum number of days that must elapse between incremental backups. This parameter is different for an archived file. A file is archived only when a client issues an archive request. The default for a backup copy group is 0 and must be cmd for an archive copy group.

Number of VERSIONS Applies only to backup files. You can specify two different parameters to tell ADSM how many versions of a backup file you want it to maintain.

The first parameter, VEREXISTS, specifies the maximum number of different backup versions the ADSM server retains for files and directories that currently exist on the ADSM client workstation. When the maximum number of versions is exceeded, ADSM rolls off the oldest version.

The second parameter, VERDELETED, specifies the maximum number of different backup versions that ADSM retains for files and directories that have been erased from the ADSM client workstation.

The default values are 2 if the file exists and 1 if the file has been deleted.

RETENTION periods Specifies how long to retain the backed up and archived files. The two retention parameters, RETEXTRA and RETONLY, for backed up files correspond to the two types of versions, VEREXISTS and VERDELETED. There is one retention parameter for archived files.

The default is 30 days for RETEXTRA and 60 days for RETONLY. The default for archived file retention is 365 days.

MODE

Specifies file backup according to whether the file has changed since the last backup. This parameter applies to incremental backups, not selective backups. The options for mode are MODIFIED and ABSOLUTE.

MODIFIED indicates that you want to back up the file only if it has changed. MODIFIED is the default value for backup copy groups. ABSOLUTE indicates that you want to back up the file regardless of whether it has changed. For archive files, the mode is always ABSOLUTE.

SERIALIZATION

Specifies how files or directories are handled if they are in use during the backup or archive process. There are four options with the serialization parameter: STATIC, SHRSTATIC, SHRDYNAMIC, and DYNAMIC.

STATIC specifies that if a file or directory is modified during the backup or archive process, ADSM will not back up or archive the file. The STATIC option is not supported on the DOS platform.

SHRSTATIC, shared static, specifies that ADSM will retry the backup operation as many times as specified in the client′s option file. The default is four retries. If the file or directory is modified during each backup or archive attempt, ADSM will not back up or archive the file.

SHRDYNAMIC, shared dynamic, specifies that if a file is modified during a backup or archive attempt, ADSM will back up or archive the file only on its last retry.

DYNAMIC specifies that even if the file is modified during the backup or archive attempt, ADSM will back up or archive the file anyway. No retries are required.

The default for both backup and archive copy groups is SHRSTATIC.

1.3.7 Function Execution

An end user who has access to the ADSM client can initiate both the backup and archive functions. These functions can also be initiated with ADSM′s central scheduling facility. Thus you can decide whether you want end users to be responsible for their backups or whether you prefer the backup process to be transparent to them.

1.3.7.1 Central Scheduling

The central scheduling facility is designed to automate both the backup and archive functions. Schedules can be established to begin on certain days or times, giving you the flexibility to manage your systems on a time schedule that makes sense for your business.

The central scheduler consists of ADSM client and server processes that cooperate to execute the scheduled functions. Therefore the ADSM client workstation must communicate with the ADSM server. If you want to automate your backups for off-hours or weekends, you need to enforce a policy that requires users to leave their workstations powered on.

The administrator is responsible for defining and maintaining the schedules and has the authority to prioritize clients so that clients that contain more important data are given preferential treatment. The administrator also can limit the number of concurrent sessions that ADSM uses.

Central scheduling is not designed for automatically restoring or retrieving files. These functions must be user initiated.

1.3.7.2 Scheduling Methods

There are two scheduling modes. The first, client polling, is started by the ADSM client. The second, server prompted, is started by the ADSM server. In both cases the backup or archive is scheduled on the basis of predefined policies. The scheduler must be started at the ADSM client before either scheduling mode can begin.

With *client polling*, an ADSM client periodically queries the ADSM server for a scheduled operation. The server sends the start date and time of the operation to the ADSM client. The client waits until that time, starts the operation, and notifies the ASDM server. On completing the operation, the ADSM client notifies the ADSM server that the operation has completed.

With *server prompted*, the ADSM client registers its TCP/IP address with the ADSM server and waits. The ADSM server connects to the client according to a predefined schedule. The ADSM client starts the operation (backup or archive) and at completion notifies the ADSM server that the operation has completed. Because server prompted scheduling is controlled by the ADSM server, it maximizes the use of scheduled sessions and other ADSM resources.

The administrator sets the schedule by specifying start time windows for those days on which either backup or archive should run. A schedule could be daily, twice a week, once a month, or whatever is appropriate for the data.

The client must start the operation within the startup window. If the client cannot perform the operation within that window (for example, because the network is down or the terminal is turned off), the client waits until the next occurrence of the startup window. Although the operation must start within the window, it may complete outside the window.

1.3.8 Client Options File

The ADSM client environment has many parameters with which you can request services, tailor the way services process, or customize the ADSM client/server system. Most of these parameters can be entered by the user, but you also can automate them in stacks of commands. For example, some parameters could be included in the workstation′s startup AUTOEXEC procedure.

These parameters also can be included in an ADSM file at both the server and client for automatic execution. The file on the ADSM server is the server options file. The file on the ADSM client is the client options file, usually called the DSM.OPT file. These option parameters can also be entered on ADSM client commands, such as those used to back up, archive, restore, or retrieve data.

1.3.8.1 Communications Options

The client options file must contain the control statements that select the communications protocols that will be used between the ADSM client and server. This is where the choice of TCP/IP, APPC, or IPX/SPX would be made. Depending on the protocols selected, other communications parameters may also be required in the options file.

1.3.8.2 Scheduling Options

The client options file also includes information that affects backup and archive scheduling. The type of scheduling to be used, client polling or server prompted, is specified along with other scheduling information including how long the ADSM client scheduler should wait between attempts to contact the ADSM server for scheduled work.

1.3.8.3 Backup and Archive Options

You can use INCLUDE and EXCLUDE statements in the client options file to control the selection of user files that will or will not be processed during backup or archive. Data compression can be selected for the ADSM client by an entry in the client options file.

If you use the DOMAIN statement in the client options file, the ADSM client can identify the volumes that should be processed during incremental backup. In a NetWare environment you can specify system information such as the bindery or directory in the client options file.

ADSM provides a command to control whether subdirectories should be included or excluded when processing their directories for backup and archive. There is also a command to control the number of times to attempt to back up or archive a file when the file is in use at the time backup or archive runs.

You can add a parameter to the client options file to control whether the ADSM client will allow overwriting files on either a restore or retrieve request. The option to have the user prompted for each file is also available.

1.3.8.4 User Interface Commands

You can specify in the client options file the format of how ADSM will display dates, time, and numbers at the ADSM client's workstations. There is also a statement that can select the national language in which ADSM messages will be displayed. All languages are not available, but this statement will be useful as additional languages are added to ADSM.

1.4 ADSM Support for NetWare

Up until this section the intent was to provide an overview of ADSM in general, with only limited comments about ADSM for NetWare. The objective of this section is to clearly identify the ADSM functions and support that are unique to the NetWare platform.

1.4.1 Supported Environments

ADSM version 1, release 2, level 01 is the level of NetWare support discussed in this book. This release of ADSM supports NetWare releases 3.11, 3.12, 4.01, and 4.02 as ADSM clients.

NetWare is currently supported as an ADSM client with the following four ADSM servers and communications protocols:

- MVS server support using TCP/IP and APPC communications protocols.
- VM server support using TCP/IP, APPC, and PWSCS communications protocols.
- AIX/6000* server support using TCP/IP, APPC, and IPX/SPX communications protocols.
- OS/2 server support using TCP/IP, APPC, and IPX/SPX communications protocols.

Note: NetWare is currently not supported as an ADSM server.

Note: To use APPC as the communications protocol you must also have an additional product, NetWare for SAA*.

1.4.2 Functions Not in NetWare ADSM Client Support

Although ADSM provides both a command line interface and a GUI to the terminal user, the NetWare ADSM client provides only a command line interface. A GUI continues to be a customer requested function and may be added in a future release of the NetWare ADSM client.

Currently there is no support for the administrative client from a NetWare ADSM client workstation. In a NetWare environment administrative functions currently need to be carried out from the ADSM server′s console or from an administrative client associated with the ADSM server. In an MVS environment this could be the TSO administrative client.

ADSM generally provides an application client to allow IBM and non-IBM products to use ADSM services in a standard callable interface. An application client is not currently available with the NetWare ADSM client but may be available in the near future in response to customer requests.

Also, as has been mentioned previously, NetWare is not supported as an ADSM server.

1.4.3 Unique NetWare ADSM Client Capability

The implementation of ADSM on NetWare offers some unique ADSM capabilities. These capabilities are a result of the unique interface between ADSM and NetWare and the unique capabilities of NetWare as a file server.

1.4.3.1 Linked NetWare Servers

NetWare allows servers to be logically linked together. This provides a way for the NetWare servers to exchange data. ADSM exploits this NetWare capability in two ways.

First, an ADSM client on one of a group of linked NetWare servers can provide full ADSM services to all of the linked NetWare servers. Thus, a single ADSM client can back up, restore, archive, and retrieve files for remote NetWare servers.

Second, if APPC or TCP/IP is used as the communications protocol for ADSM, a single NetWare server could act as a gateway to the ADSM server for the other linked NetWare servers. This capability could be particularly useful with APPC because it saves the cost of an additional product on the nongateway NetWare servers.

1.4.3.2 NetWare′s SMS Interface

ADSM backups are done using NetWare′s storage management services (SMS) interface. This interface allows ADSM and NetWare to take full advantage of storage management function placed in either of the products.

Use of NetWare′s SMS interface enhances data security for ADSM functions. Full NetWare data security is available through ADSM including the automatic backup of security information for a file as ADSM backs up the file.

ADSM′s use of the SMS interface also means that backup/restore and archive/retrieve provide full support for NetWare′s bindery and directory. These NetWare components are defined and discussed in Chapter 2.

Chapter 2. Introduction to NetWare

This chapter introduces the NetWare operating system to those people who are familiar with ADSM but need to understand NetWare.

The chapter provides sufficient background information to enable you to implement the NetWare ADSM client. Emphasis is on those features of NetWare that are relevant for someone implementing ADSM on NetWare for the first time.

NetWare is a network operating system designed to provide shared resources for a number of workstation users. A NetWare system consists of a central NetWare server and a number of workstation users connected together through a LAN.

The user workstations can be either DOS, OS/2, UNIX, Macintosh, or Windows. Of these, DOS and Windows are by far the most common today. The central NetWare server machine provides the following shared resources for the workstation users:

- File serving
- Print serving
- Application serving.

NetWare was first introduced in 1983. It started out as an operating system that ran on a proprietary Motorola 68000 computer. With the introduction of the IBM PWS, Novell developed NetWare to run on Intel-based PWSs. Since then NetWare has gone through many different versions. Today there are two primary versions, NetWare version 3 and NetWare version 4.

Versions 3 and 4 account for the vast majority of installed NetWare systems. ADSM supports both NetWare versions 3 and 4. In the sections that follow we introduce NetWare versions 3 and 4 and Netware for SAA, the product that provides SNA connectivity for NetWare servers. If you are already experienced with NetWare you may want to skip to Chapter 3.

2.1 NetWare Versions 3.11 and 3.12

In this section we cover the architecture of NetWare versions 3.11 and 3.12, ADSM′s currently supported levels.

2.1.1 Server Architecture

NetWare version 3.1x is a multitasking network operating system that runs on Intel 80386-based PWSs and above. NetWare is not a native operating system that can be booted from a disk. It requires DOS to be loaded first and then uses DOS to load the NetWare kernel. This is in distinct contrast to other PWS operating systems such as DOS and OS/2, which are native operating systems.

A NetWare server may be booted from a floppy disk or from a small DOS partition on the disk from which the system boots. This DOS partition contains a

DOS system and the NetWare system code required to start the server. In addition to the DOS system the following minimum files are required to start NetWare:

1. SERVER.EXE

   This is the base NetWare operating system file that is executed when NetWare is started from DOS. Once executed it takes over control of the processor from DOS. Typically a statement is added to the DOS AUTOEXEC.BAT to execute this file.

2. STARTUP.NCF

   When SERVER.EXE is executed the first file it processes is its STARTUP.NCF. NetWare command files (NCFs) are similar to DOS batch or OS/2 command files and are used to execute NetWare commands. The STARTUP.NCF file for NetWare performs a similar function to the CONFIG.SYS file on OS/2. It is used to load a disk driver that enables NetWare to access its NetWare volumes on the server disk to continue the server startup process.

3. A NetWare disk driver

   This is the disk driver loaded by the STARTUP.NCF file. The SERVER.EXE file uses the driver to access the NetWare volumes defined on the server disk.

When SERVER.EXE has executed and loaded the required disk driver, it mounts its first NetWare volume, SYS:, and continues the startup process. SYS: is the NetWare logical volume that resides in a NetWare partition on the server disk. It contains all of the additional server executable modules that go to make up a NetWare server. At this point the system has become a NetWare server. The DOS system used to start the process is no longer required or used, although it is still loaded in the processor memory.

When NetWare has successfully mounted its SYS: volume, it executes a file called AUTOEXEC.NCF, which resides in the SYS:SYSTEM directory. This NetWare command file is used to complete the startup and configuration process of the NetWare server.

2.1.2 Loadable Modules

A NetWare server is initially started with a minimal DOS system. Control is then passed to the SERVER.EXE file, and NetWare takes over control and continues its startup process. The SERVER.EXE file is the basis of the NetWare kernel. It provides the base operating system functions, such as multitasking and memory management.

The NetWare internal architecture enables the base operating system to be dynamically reconfigured by the addition of system extensions called NetWare loadable modules (NLMs). NLMs can be dynamically loaded or unloaded allowing the configuration of the NetWare server to be changed. The SERVER.EXE file can be considered as a software bus into which NLMs are plugged or from which NLMs are unplugged. Figure 10 on page 25 illustrates this concept of a software bus with a variety of NLMs loaded.

When an NLM is loaded it becomes an integral part of the NetWare operating system and is permanently loaded in server memory. At server initialization

time, after the SYS: volume has been mounted, the AUTOEXEC.NCF file is used to load the additional NLMs for the required server configuration.

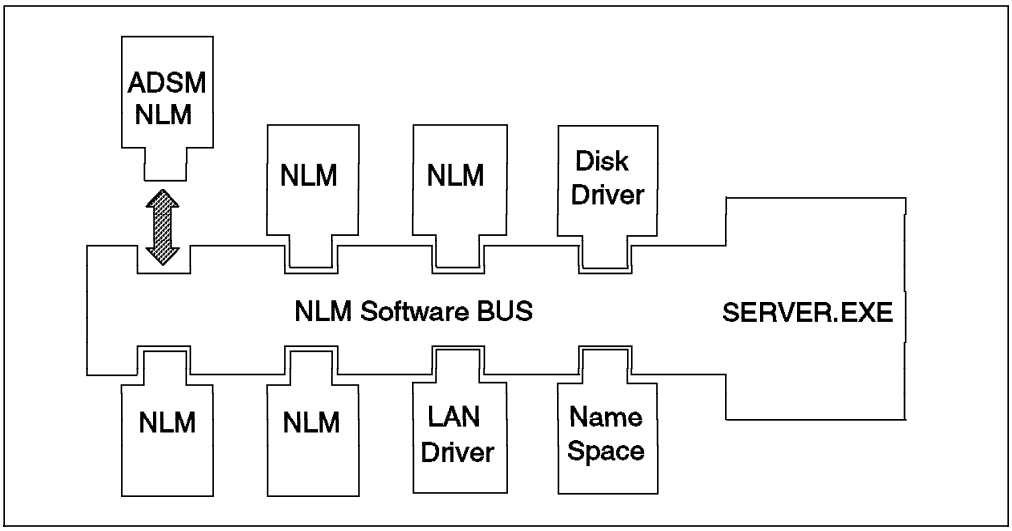


Figure 10. SERVER.EXE and the NLM Software Bus

There are four types of NLMs:

- Disk drivers

  Disk drivers provide specific device drivers for different disk drives. They have a file extension of DSK. For example, PS2SCSI.DSK is the NLM device driver for the PS/2 SCSI disk adapter.

- LAN drivers

  LAN drivers provide drivers for the various LAN adapter cards. They have a file extension of LAN. For example, TOKEN.LAN is the NLM device driver for an IBM PS/2 token-ring card.

- Name space modules

  Name space modules provide support for other file system types to allow them to store data on a NetWare server. They have an extension of NAM. For example, OS2.NAM is the name space module for the OS/2 HPFS file system. This would have to be loaded if a user's OS/2 workstation wanted to store HPFS files on a NetWare server.

- General utilities and applications

  These are the general purpose modules that provide system functions and applications. They have a file extension of NLM. For example CLIB.NLM is the NetWare C run-time library.

NLMs are loaded using the LOAD command and unloaded using the UNLOAD command at the NetWare server console. Alternatively, they can be loaded from within a .NCF batch file. Here is an extract from an AUTOEXEC.NCF that is used to configure a server:

```
.
LOAD TOKEN
BIND IPX TO TOKEN NET=2
LOAD STREAMS
LOAD CLIB
SEARCH ADD SYS:ADSM
LOAD DSMC
.
```

In this example the TOKEN LAN driver is loaded and the NetWare IPX protocol is bound to it. Then the STREAMS and CLIB NLMs are loaded. Note that it is not necessary to specify the full name and extension of the NLM or its location. By default NLMs reside in the SYS:SYSTEM directory. This is where NetWare looks for them when a load command is executed.

If NLMs are located in other directories, NetWare must be told where to look for them with a SEARCH statement. In the example above, a search path is added for the SYS:ADSM directory before DSMC is loaded. DSMC.NLM is the ADSM command line client for NetWare.

When loaded, NLMs become part of the NetWare operating system. This fact potentially could allow a poorly designed NLM to adversely affect the system as a whole. Novell has a certification process to test NLMs to ensure that they behave properly when loaded. Satisfactory completion of these tests obtain for the NLM the Novell “tested and approved” label. The ADSM client for NetWare has been tested and certified.

2.1.3 File System

The NetWare file system differs from other PWS-based operating systems such as OS/2 or DOS. The file systems on those systems are hierarchical and consist of drives, directories, and files. For example, there is normally a C: drive with a number of directories, subdirectories, and files below it. A drive is usually a physical disk or a partition of a disk. The NetWare file system differs from this. It is organized in a logical manner into volumes, directories, and files.

2.1.3.1 Volumes

A NetWare partition on a disk is an area of the disk that has been formatted by NetWare for use as a NetWare logical volume. A NetWare volume can be mapped to a single NetWare partition or across multiple partitions on many different physical disks. Figure 11 on page 27 illustrates a possible NetWare file system configuration.

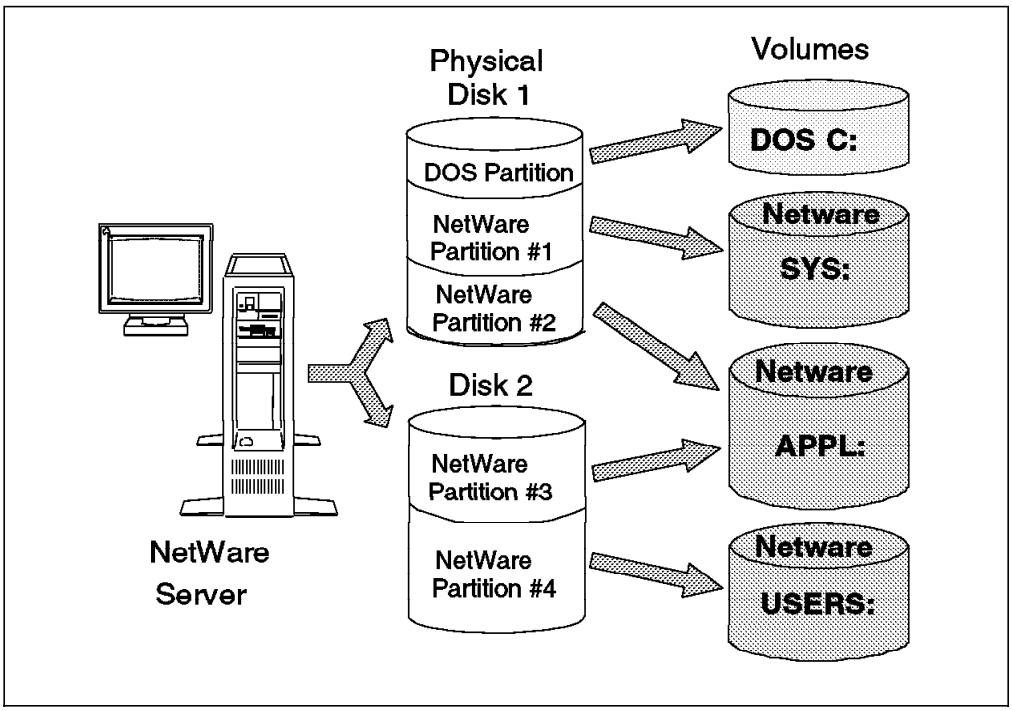


Figure 11. NetWare File System Overview

In this example the NetWare server has two physical disks. The first disk is partitioned into a DOS partition and two NetWare partitions. The DOS C: drive is used to start the server initially. The first NetWare partition is assigned as the SYS: volume. Partitions 2 and 3 are assigned as a volume called APPL:. Partition 4 is assigned as a volume called USERS:.

A NetWare volume is a logical entity that can span multiple physical disk partitions. This is called *volume spanning* and a volume can consist of up to 32 segments, or partitions. NetWare version 3.1x supports a maximum of 64 logical volumes.

The SYS: volume is created as part of the server install process and must always be mounted.

2.1.3.2 Directories and Files

A NetWare volume is divided in a hierarchical manner similar to DOS and OS/2. Below a volume there are directories, subdirectories, and files. To specify a file on a particular server the fully qualified path name takes the following form:

SERVERNAME/VOLUME:DIRECTORY/SUBDIRECTORY/FILENAME

Note: In the example above the separator character between entries is a slash (/). NetWare allows either slashes (/) or back slashes (\) to be used.

This ability to specify server name, volume, and file path is very flexible. However it is of little direct use to users of DOS or OS/2 workstations. Neither DOS nor OS/2 understands what a NetWare volume is. They use drive letters such as C: or D: to identify the root of a file system structure.

To facilitate the use of NetWare file systems, NetWare provides the ability to map a drive. A DOS workstation user logged into a NetWare server can use the

NetWare MAP command to associate a drive letter with a point in the server's file system, for example:

```
MAP H:=ITSC-NW1/SYS:USERS/TIM
```

This command maps the DOS drive H: to the USERS/TIM directory on the SYS: volume on a NetWare server called ITSC-NW1. The user could then change to directory H: and have access to this point in the NetWare file system.

When a NetWare server is first installed, a standard directory structure is created. This structure consists of the SYS: volume and five basic directories as seen in Figure 12.

The SYSTEM directory contains the NetWare system files including all of the NLMs supplied with NetWare. The PUBLIC directory contains the executable NetWare utilities. The LOGIN directory contains the user LOGIN utilities. The MAIL directory contains each user's LOGIN scripts. The ETC directory contains sample configuration files for use with TCP/IP.

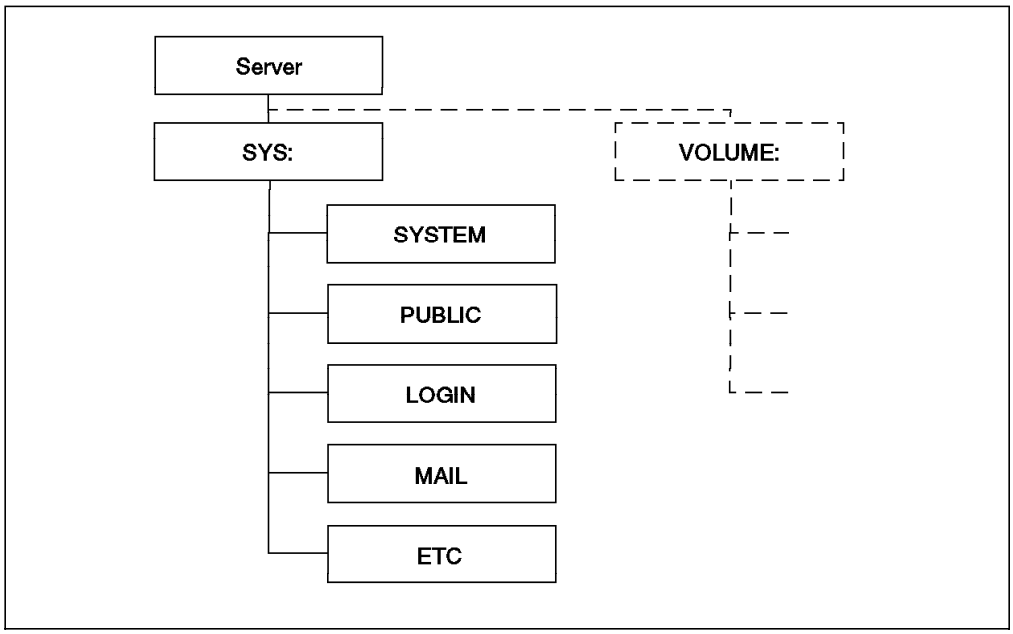


Figure 12. NetWare Standard Directory Structure

This standard directory structure must always exist on a NetWare server. Once installed, it can be extended by either adding additional directories or defining additional volumes.

2.1.3.3 Name Spaces

The NetWare file system is a Novell proprietary file system. Files appear as DOS file allocation table (FAT) format files. However, the underlying file system actually stores them in its own proprietary format. This ability to emulate a file system provides the capability to support otherwise incompatible file systems on a single NetWare server.

This capability is called *alternate name spaces*. Name spaces are NLMs that can be loaded to provide support for file systems such as the OS/2 HPFS or Macintosh file system. When these name spaces are loaded, users can store

files in those formats on the NetWare server. The files appear to the user in the correct format, but they are stored in the NetWare file system in Novell's own format.

2.1.4 Security

With any form of operating system where multiple users have access to shared resources, security is required. NetWare is no different and implements security at two levels, server access security and file system security.

2.1.4.1 Server Access Security

Access to a NetWare server is controlled by userids, passwords, group definitions, and other restrictions such as disk usage and group membership. These controls are maintained in a special NetWare security database called the *bindery*.

The bindery comprises three separate types of entries:

- Objects

  Physical or logical objects are defined on a server. An example of an object is a userid.

- Properties

  Every object defined on a server can have a number of properties. For a userid these could include the user's name, password, and account restrictions.

- Values

  Values are the actual values defined for the properties of an object. An example would be the user's password that has to be entered

The bindery is a critical system resource on a NetWare version 3.1x server. All definitions for the server and all of its entities are held in the bindery. The bindery consists of the following three hidden files in the SYS:SYSTEM directory:

- NET$OBJ.SYS for the bindery object definitions
- NET$PROP.SYS for the bindery object properties
- NET$VAL.SYS for the values of the object definitions.

When a NetWare server is started, one of the first things that happens is that the bindery is opened. It remains open all the time the server is active. It is only closed when the server is shut down.

The integrity of the three open bindery files is absolutely critical. If the bindery is corrupted and cannot be recovered, the integrity of the NetWare server is compromised. NetWare provides some basic utilities to fix bindery problems. However, these utilities are no substitute for regular, consistent backups of the bindery.

2.1.4.2 File System Security

Once a user has logged in to the server, some mechanism is required to control access to the file system. NetWare implements file system security separately from the basic server access security.

There are two distinct parts to NetWare file system security, rights access and file attributes. Rights access is the ability to control that portion of the NetWare file system to which a user or a group of users has access, or *rights*. NetWare can assign rights to allow users to perform functions such as read, write, create, and modify at the directory or individual file level. A user or group of users with rights to a directory or file is called a *trustee*, and the rights assigned are called *trustee rights* or *trustee assignments*.

Figure 13 illustrates how rights access works. In this example a number of users are defined as a group called STAFF. These users require read, write, create, and erase privileges for the SYS:ITSC directory and all directories and files below it.

If a trustee assignment is made giving the STAFF group rights to that directory, they will also apply by default to all directories and files below that point. NetWare trustee rights flow down from the point in the file system at which they are assigned.

All members of the STAFF group automatically have the same trustee rights as assigned to the group. As users are added to, or removed from, the STAFF group they automatically gain or lose the associated rights.

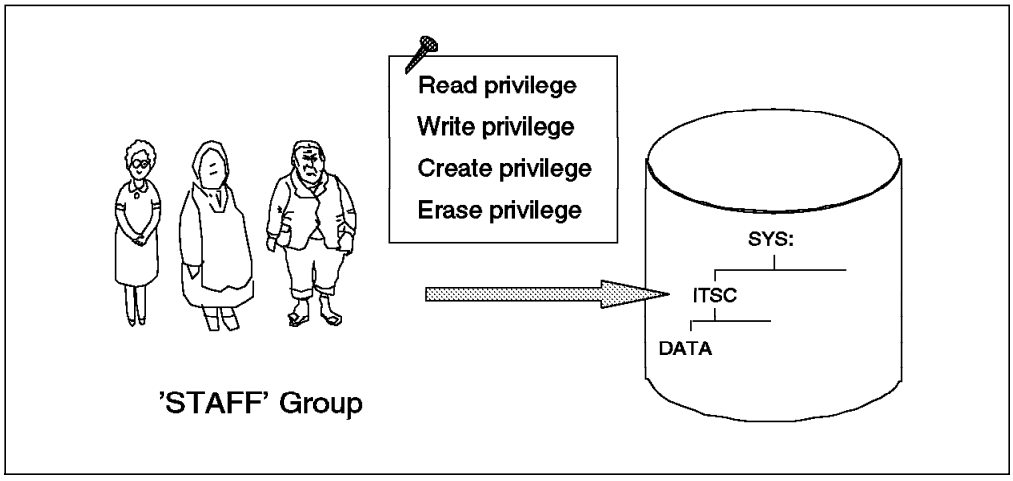


Figure 13. NetWare Trustee Rights Assignments

The assignment of rights that flow down from a directory through the subsequent file structure is very powerful. However, it is often necessary to limit assigned rights for certain directories and files. This could be achieved by assigning different trustee rights for those directories and files.

NetWare provides another method of limiting assigned rights, the inherited rights mask (IRM). The IRM is a filter that blocks rights from flowing down from parent directories. It is applied to all directories and files.

In the example above, it might be desirable to prevent users from erasing files in the SYS:ITSC/DATA directory. The IRM for the DATA directory could be modified to block the erase right from flowing down from the ITSC directory. NetWare supervisors set trustee assignments and the IRM using a utility such as FILER or SYSCON. Trustee assignments and IRMs are stored within the file system itself, not in the bindery.

NetWare file system security can assign directory and file attributes. These are similar to the DOS file attributes but more comprehensive in capability. Attributes such as copy inhibit, delete inhibit, and execute-only provide significantly more flexibility than DOS′s archive, system, and hidden attributes.

These NetWare file attributes can sometimes conflict with trustee rights. Where applicable, attributes override trustee rights. As with trustee rights, attributes are stored within the file system itself. NetWare supervisors can assign attributes using utilities such as FLAG and FLAGDIR.

2.1.5 Users

In this section we look at the types of NetWare users and the limitations placed on them. When a NetWare version 3.1x server is installed, two users are automatically created, GUEST and SUPERVISOR. These represent the two extremes of NetWare users. GUEST is exactly what the name implies, a guest ID that requires no password to log in to the server. SUPERVISOR is the default NetWare administrator ID and has authority to do anything, to anybody on the NetWare server. This is an extremely powerful ID that NetWare administrators heavily guard.

After installation of the NetWare server, the administrator defines and sets up users that run applications, store data, and use print servers for shared printers on the NetWare server.

Users can be linked together into groups for administration purposes. In general these users or groups have only limited authority on the NetWare server. They might be given trustee rights to their data on the file system, but they have limited ability to perform other tasks and are often limited in the amount of NetWare server resource, such as disk space, they can use.

Administration of a NetWare server is performed by two special types of users: supervisors and work group managers. Supervisors are users who have been given supervisory rights by the original SUPERVISOR ID. They are given what is known as security equivalence to SUPERVISOR and have exactly the same power as the original SUPERVISOR ID. This is a way of delegating supervisor responsibility to other users having to use the original SUPERVISOR ID.

Supervisors can also define what is known as work group managers. These are users who have a subset of the supervisor′s rights, but only for a group of users. Work group managers can administer their own groups of users, performing tasks such as creating and deleting users.

2.1.6 User Interface

The NetWare user, whether a typical user or a supervisor, requires some means of accessing the NetWare server. There are two ways of accessing the server: through the server′s or the NetWare requester′s console, as shown in Figure 14.

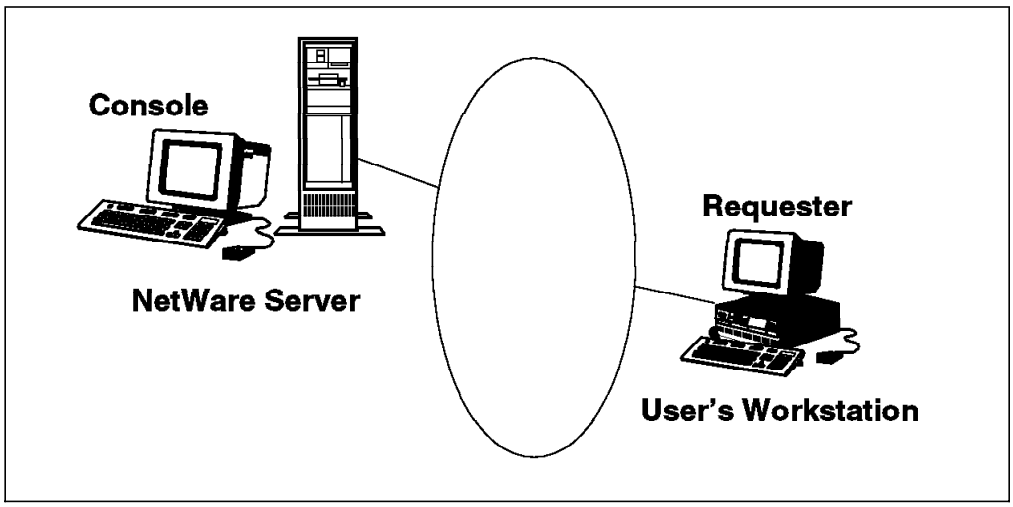


Figure 14. NetWare User Interfaces

2.1.6.1 Server Console

NetWare has as its primary interface a console. This is the physical screen and keyboard attached to the NetWare server. It is a text-based interface and dedicated in its use as the console. At the console NLMs can be loaded or unloaded, NetWare volumes can be mounted, and the server shut down and restarted. The console is the only place where NLMs can be manually loaded. If an NCF file is used to load a series of NLMs, again it can be run only from the server console.

The NetWare server supports multiple logical console sessions. If a NetWare utility, such as MONITOR, is loaded at the console, it will display its own logical console. It is possible to have many utilities running, all with their own logical consoles. These consoles can be accessed by pressing <CTL> and <ESC>, then selecting the required console session. The ADSM NLM, DSMC, provides its logical console in this way. The NetWare console is currently the only way of using the ADSM NetWare client.

2.1.6.2 Requester

The second and more usual way of accessing the server is through a user′s workstation connected to the server by means of a LAN. This user workstation runs a piece of software, the NetWare requester, which enables the communication between user workstations and the NetWare server. A user can connect to a server and use its resources, while retaining the usual workstation user interface for DOS, OS/2, Macintosh, or Windows.

When users log in to a NetWare server through a requester, NetWare processes a LOGIN script to enable them to gain access to their authorized resources. This sets up default drive maps. Users can then use the MAP command to alter the DOS or OS/2 drive letters to locations within the NetWare file system, as explained in 2.1.3.2, “Directories and Files” on page 27.

Once users are logged in to a server, a variety of NetWare commands and utilities are available them depending on their authority. The commands and utilities are executable programs that can be run from a DOS or OS/2 prompt, rather than NLMs that have to be loaded at the console. Examples are FILER for managing files and directories and SYSCON for managing users and groups.

NetWare also has a remote console facility called RCONSOLE. This utility can be run at a requester and provides the same console interface as the physical NetWare console. All functions that need to be performed at the console can be performed from an RCONSOLE session, including shutting down the server.

To use RCONSOLE a user must have supervisor equivalent rights or must know the RCONSOLE password. The REMOTE.NLM and RSPX.NLM must also be loaded at the NetWare server. RCONSOLE enables large networks of servers to be managed from a central point, without requiring physical access to the individual servers.

2.1.7 Connectivity

As a network operating system NetWare depends on connectivity. NetWare connectivity can be logically divided into two categories:

- NetWare server to requester connectivity (LAN based)
- NetWare server wide area connectivity.

Figure 15 illustrates the main connectivity protocols discussed in this section.

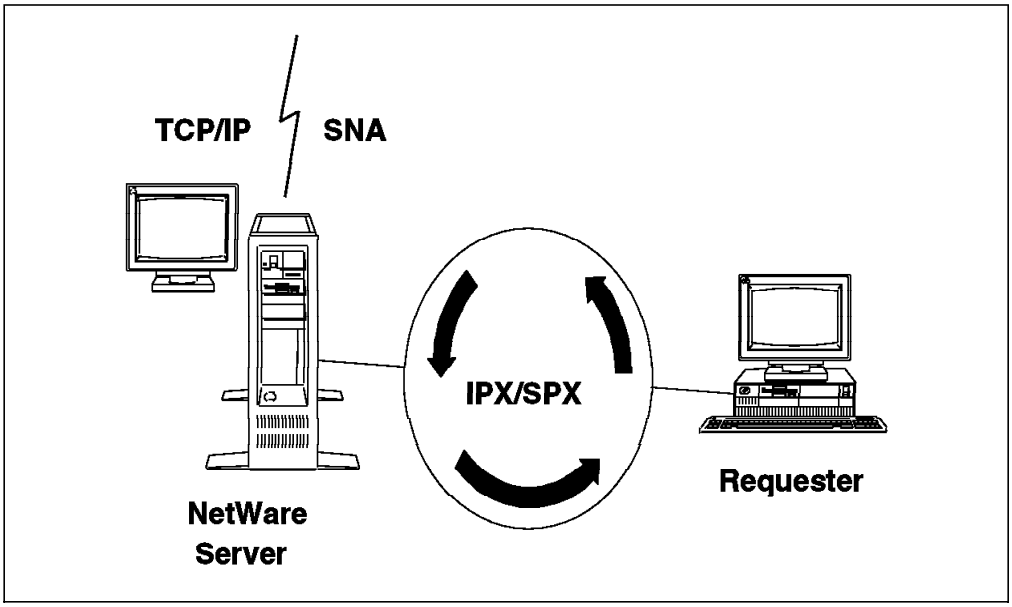


Figure 15. NetWare Connectivity

The main NetWare protocol used between NetWare servers and requesters is IPX/SPX. Internetwork Packet Exchange (IPX) is a protocol similar to IP in TCP/IP. IPX is a connectionless protocol, where packets are sent, but a response is not required. With IPX there is no guarantee that the packet will be received.

Clearly a connectionless protocol such as IPX on its own is not sufficient for application communications support, so NetWare implements the Sequenced Packet Exchange (SPX) protocol on top of IPX. SPX is a connection-based protocol where delivery of the packet is guaranteed, and error handling is performed. IPX/SPX is the main suite of protocols used with a NetWare system. However, IPX/SPX is not always supported on other systems; for example, VM and MVS do not support it.

If connectivity is required between NetWare servers and VM or MVS systems, additional protocols need to be implemented. For our purposes the two main protocols are TCP/IP and SNA.

NetWare provides as part of the base operating system a TCP/IP protocol stack, that is, a collection of NLMs that provide the base TCP/IP services. However, applications such as Telnet, FTP, or NFS are not provided. These must be purchased separately.

SNA connectivity is not supplied with NetWare. If SNA is required, an additional product, Netware for SAA, must be purchased (see 2.3, “NetWare for SAA” on page 40).

NetWare supports multiple protocols running on the same LAN and adapter card. A NetWare server could have a single token-ring adapter that can be used simultaneously for IPX/SPX, TCP/IP, and SNA traffic.

2.1.8 Availability Features

NetWare version 3.1x has a number of availability features that provide varying degrees of data and server protection. Novell has defined a system fault tolerant (SFT) architecture. SFT is similar to the industry standard RAID architecture. There are three levels of SFT:

- SFT I

  SFT I is also known as *hot fix*. It is a read after write data verification process. When NetWare writes a block of data, it immediately reads it to verify what was written. If the data read is different from the data written, NetWare assumes that the sector on the disk is defective and rewrites the data to a special area on the disk known as the hot fix redirection area. The bad sector on the disk is recorded and not used again.

  The hot fix area is created when the NetWare partition is formatted and is typically 2% of the partition capacity. It is similar in principal to alternate tracks on mainframe DASD such as 3390s. SFT I is implemented by default on all NetWare 3.1x servers.

- SFT II

  SFT II implements disk mirroring or disk duplexing. Disk mirroring is where two physical disks are attached to the same disk controller on the NetWare server. Any data written to one disk is mirrored on the other. Disk

duplexing is slightly different in that separate disk controller cards are used for each disk, thus removing the single point of failure that exists for mirroring. In both disk mirroring and disk duplexing the disks are copied at the physical level. All NetWare partitions and volumes on the disk are mirrored. SFT II is part of the base NetWare 3.1x system, but its use is optional.

- SFT III.

  SFT III implements server level mirroring, that is, a second NetWare server acts as a hot standby. The second is an exact copy of the primary server. Whatever actions are performed on the primary server are replicated on the second server. The two servers are connected together by means of a high-speed bus, normally using an FDDI connection.

  SFT III is not part of the base NetWare 3.1x system. If customers want to implement SFT III, they must purchase the separate NetWare SFT III product. This product is based on NetWare 3.1x with the additional functions to provide the server-to-server connectivity.

NetWare also provides a transaction tracking feature called the transaction tracking system (TTS). TTS enables file and database I/O operations to be backed out in the case of a system failure. For example, if a large file is being updated and the server goes down in the middle of the operation, the incomplete transaction will be backed out when the NetWare server is restarted.

2.1.9 Storage Management Services

Novell introduced storage management services (SMS) with NetWare version 3.1x. SMS defines a set of standards that provide platform-independent storage management services. It provides an application programming interface (API) that software vendors can use to produce storage management products. Products written to the SMS API are isolated from the underlying NetWare operating system. Therefore, as Novell makes changes to NetWare, those products are not affected.

Additionally the SMS API enables storage management applications to access objects such as NetWare′s bindery and file system in a consistent manner. These objects are built to a proprietary NetWare architecture and contain essential information that must be managed properly. Such management can be achieved only by using an SMS product. An SMS product should, for example, close the bindery before backing it up and then reopen it automatically. When backing up file systems, it should also back up all trustee rights and file attributes.

The SMS architecture is implemented as a set of building blocks based on NLMs. Figure 16 on page 36 illustrates these building blocks and where they might physically reside on NetWare servers. The specific blocks are defined in the next sections.

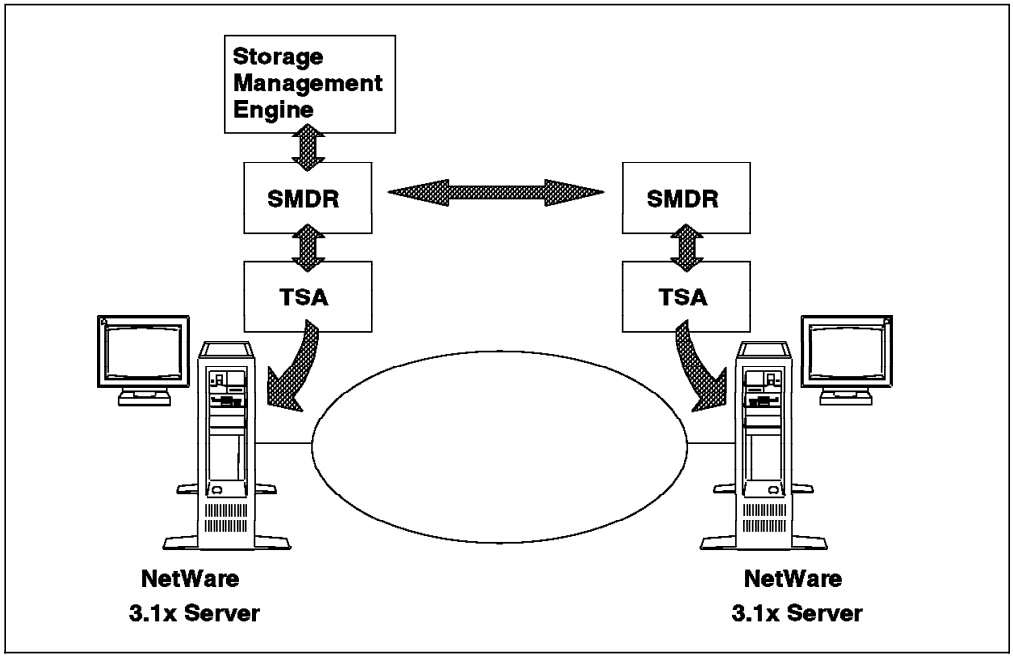


Figure 16. NetWare Server SMS Components

2.1.9.1 Storage Management Engine

The storage management engine (SME) is the SMS term for the storage management application on the NetWare server. SME is the application that initiates backups and restores of data. As an application it is isolated from the underlying operating system. When it requires data from a server or needs to put data back, the SME does so by passing a request to the storage manager data requester (SMDR). The ADSM NetWare client is an SME. It runs as an NLM and backs up and restores data on a server by passing requests to the SMDR.

2.1.9.2 Storage Management Data Requester

SMDR is SMS′s communications mechanism between SME applications such as ADSM and the data they are managing. The ADSM client provides an SMDR NLM. The SMDR function is the communications vehicle for SMS. However, it does not do the physical reading and writing of data, which is performed by a target service agent (TSA) on the NetWare server.

The SMDR enables ADSM to communicate with local as well as remote servers. In this way a single ADSM NetWare client running on a server can manage data on its own physical machine and on other remote NetWare servers that have an SMDR NLM. Communication between SMDRs uses the standard NetWare protocol, IPX/SPX. No additional configuration is required to implement local and remote NetWare server communication, other than loading the required SMDR NLMs. The server name is used to identify which servers are to be accessed.

2.1.9.3 Target Service Agent

The Target Service Agent is the piece of code that performs the specific data requests on the NetWare server. The TSA understands the format of the file system and how to access it. ADSM provides TSA NLMs for NetWare version 3.11 and 3.12 servers, called, respectively, TSA311.NLM and TSA312.NLM. These TSAs understand how to:

- Automatically close the bindery before it is backed up and then reopen it afterward
- Access NetWare volumes
- Back up and restore NetWare file system attributes and trustee rights assignments.

With NetWare version 3.1x, a single TSA is required for a server. NetWare version 4 introduces additional functions and TSAs. These are covered in more detail in 2.2.3, “Changes to SMS” on page 39.

2.2 NetWare Version 4

This section highlights the differences between NetWare versions 3.1x and 4 that are relevant to ADSM.

NetWare version 3.1x is based on the concept of a single server and its users. A NetWare version 3.1x server is basically a stand-alone, self-contained system. Users can log in or attach to multiple servers concurrently, but a NetWare server is generally ignorant of other NetWare servers.

With NetWare version 4, Novell introduced the concept of an enterprisewide system rather than a series of independent server systems. NetWare version 4 servers can be grouped together in a domain concept, similar to the way in which an OS/2 LAN Server implements domains. Users defined on a NetWare version 4 server can gain access to any resource on the network without having to be specifically defined to them in all NetWare systems.

The major change in NetWare version 4 is NetWare directory services (NDS) as a replacement for the bindery. Additionally the file systems have been enhanced significant and new SMS capabilities have been added.

2.2.1 Directory Services

NDS is a global, distributed database that maintains all information about every resource on the network. It can be replicated across multiple servers. NDS replaces the bindery used by NetWare version 3.1x.

NDS organizes resources in a hierarchical structure, independent of their physical location. Users can log in to the network and access resources without needing to know to which server to LOGIN.

All resources that are defined in the NDS are *objects*. These objects could be users, servers, volumes, or printers and are defined somewhere in the NDS tree structure. Within the NDS tree there are two types of objects: *leaf objects*, which define single things like users, and *container objects*, which contain a collection of leaf objects. Figure 17 shows a sample NDS tree structure.

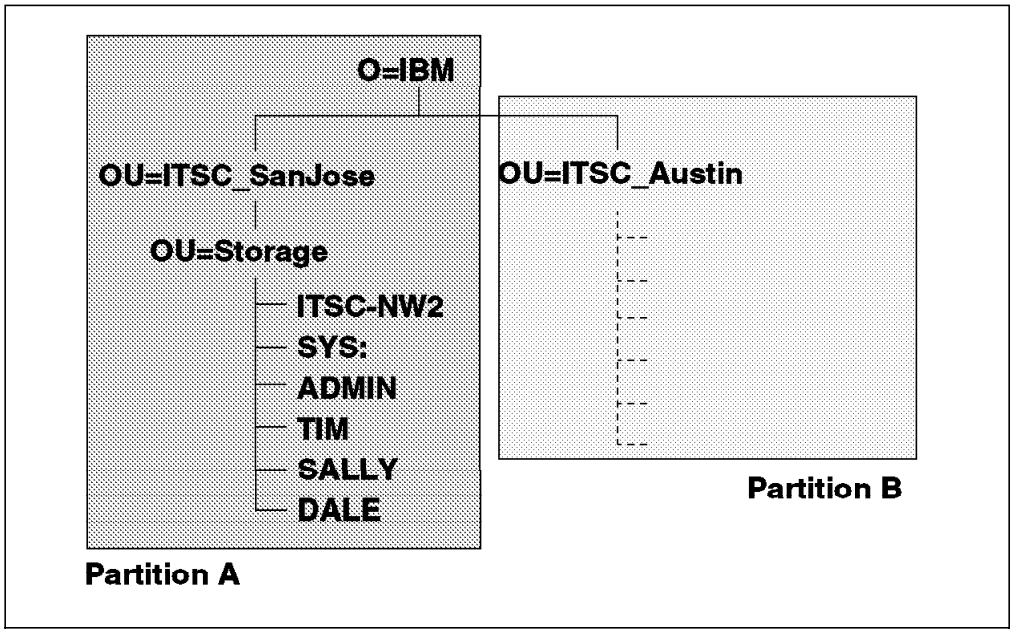


Figure 17. NetWare Directory Services Tree Structure

In this example there is an organization (O) called IBM. Below that are two organization units (OUs), ITSC San Jose and ITSC Austin. Below ITSC San Jose is another OU of storage where the leaf objects are defined. These leaf objects are called common names (CNs). Under storage there is a server called ITSC-NW2, a volume called SYS:, and a number of users. All of these objects can be given trustee rights that permit them to perform tasks or access resources in the network.

NDS is a very flexible and powerful system that enables a logical representation of an enterprise to be defined. However, NDS can potentially become very large and thus have performance and maintenance implications. Also the NDS database can be replicated physically onto many different NetWare servers, thus affecting availability and performance.

If a user wants to log in to the ITSC-NW2 server, it does not make sense to do the authentication many miles away in Austin. Portions of the NDS database can be partitioned and stored locally. In the example above, partition A of the NDS is created and held locally on the San Jose NetWare server. It can then be replicated on a remote server for availability purposes. Partition B could be held on a server in Austin and replicated for availability.

2.2.2 File System

With NetWare version 4, Novel introduced new functions on the NetWare file system, including file compression and data migration. File compression compresses files that have not been accessed for a certain period of time. Directories or files can be flagged so that they are compressed after a specific

period of time without access. The compressed files remain on the file system and are decompressed automatically when a user accesses them. File compression is enabled by default when NetWare version 4 is installed.

ADSM running on a NetWare version 4 server can back up files that have been compressed by NetWare without decompressing them. When restored, the files are stored in compressed format. NetWare version 4 has a new file attribute for compressed files. As with other file attributes, this new attribute is backed up and restored correctly.

Data migration is a feature that migrates files that have not been accessed for a specific amount of time to other media. When users access the migrated files, they are automatically recalled to disk. Migration is disabled by default. It must be configured if required. ADSM backs up migrated files directly from the migration device without staging them to disk first.

2.2.3 Changes to SMS

With NetWare version 4, Novell introduced some new SMS API features. First, the introduction of NDS requires a new TSA. ADSM provides a TSANDS.NLM for backing up the NDS on a NetWare version 4 server. Second, the capability was announced to back up data resident on users′ DOS and OS/2 workstations. To enable this a number of new components were added to the SMS architecture. Figure 18 illustrates the new components that are used to back up user workstations.

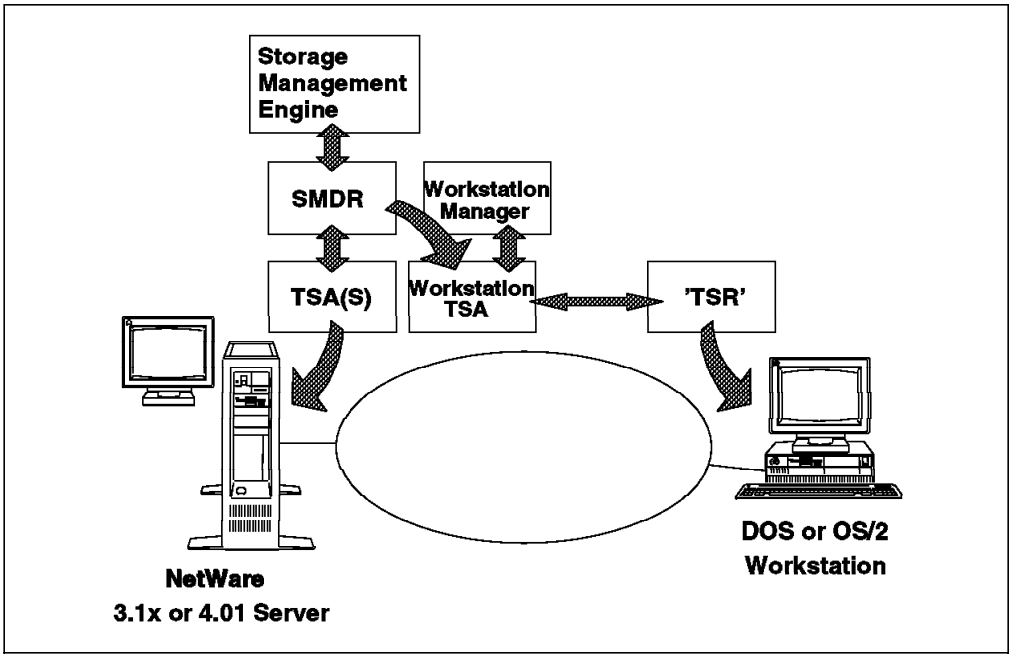


Figure 18. Changes to SMS Function with NetWare Version 4

This capability requires new TSA support for DOS and OS/2 requesters. There are two new TSA NLMs that provide support for NetWare version 4 requesters, TSA_DOS and TSA_OS2. These are loaded on the NetWare server and are used for communicating with the workstations on behalf of the SMDR. At the workstation a TSA.EXE program is executed. This is a terminate and stay

resident (TSR) program that sits in memory waiting to be accessed from the TSA on the NetWare server.

The user workstations do not need to be logged in to a NetWare server for the TSR to be used. They simply need to be powered on the TSR program executing. The TSA.EXE program is configured with a name for identification.

Another feature of the NetWare server is the workstation manager NLM, which is responsible for keeping a list of workstation names available for backup.

These features have been applied to all NetWare versions that support the SMS standards; namely, versions 3.11, 3.12, 4.01 and 4.02. The initial release of NetWare version 4, 4.0, did not support SMS. ADSM does not support NetWare version 4.0.

2.3 NetWare for SAA

NetWare has no native SNA connectivity protocols. NetWare for SAA can be used to provide connectivity. NetWare for SAA is the product that provides this function. NetWare for SAA runs on either a NetWare version 3.1x or 4.x server and provides SNA connectivity for the server itself and the requester workstations.

NetWare for SAA can be configured in two different ways:

- As an SNA physical unit (PU) type 2.0

  Provides SNA logical unit (LU) type 0, 1, 2, 3 and dependent 6.2 support.

- As an SNA PU type 2.1.

  Provides independent LU type 6.2 support only. PU type 2.1 is the required configuration for ADSM′s implementation of APPC.

NetWare for SAA supports connections to hosts that use token-ring, Ethernet, SDLC, X25, and direct channel attachment.

2.3.1 Packaging

NetWare for SAA is sold in packages that support varying numbers of host sessions. Packages of 16, 64, 128, or 254 sessions are available. For each version you get an equal number of dependent (PU 2.0) and independent (PU 2.1) sessions.

With NetWare for SAA a maximum of two host connections, known as service profiles, can be defined on a NetWare server. These service profiles define the PU type 2.0 or 2.1 connections to the host. If one PU type 2.1 service profile is defined for independent LU type 6.2 sessions, only one PU type 2.0 can then be defined for LU type 2 3270 and other dependent sessions.

2.3.2 Run-time Versions

NetWare for SAA can be installed on NetWare version 3.1x or 4.x servers. However, on a busy file server, this might not be desirable for performance reasons. A significant number of customers choose to run NetWare for SAA on a separate NetWare server acting as a communication gateway for their file servers and users.

Rather than expecting customers to go out and buy a NetWare server license for this, Novell includes a special license with NetWare for SAA. This license is known as a *run-time version*. It is a full function version of NetWare but with the number of allowed connections from requesters limited to 4. This number is sufficient to enable administrators adequate access to the server for setup and configuration purpose, but not enough to realistically use it as a true file server. Run-time versions of NetWare version 3.12, 4.01, and 4.02 are currently supplied with NetWare for SAA.

Chapter 3. Implementing the ADSM NetWare Client

In this chapter we discuss the implementation of the NetWare ADSM client. The first section covers installation of the ADSM client code and tailoring the basic client options. Subsequent sections look at the setup of the various connectivity options between the ADSM NetWare client and an ADSM server. There are separate sections on:

- IPX/SPX connectivity
- TCP/IP connectivity
- APPC connectivity.

VM PWSCS, the other supported communications protocol for the ADSM NetWare client, is not covered in this book. It is supported only for the NetWare version 3.11 ADSM client and VM ADSM server.

In the last section of this chapter we explain how to start the client for the first time. Throughout this book's examples the release 2 ADSM client is used. This ADSM release supports NetWare versions 3.11, 3.12, 4.01, and 4.02. It is assumed that the ADSM server is already installed and operational.

3.1 Installing the ADSM Client

The ADSM NetWare client is supplied on a single 3.5 inch diskette. The diskette contains all of the required NLMs for NetWare versions 3.11, 3.12, 4.01, and 4.02. Installation is initially performed from the NetWare server console and requires access to the diskette drive on the server PWS. A remote console session could be used, but the diskette must be in the NetWare server diskette drive.

The diskette contains an ADSM installation NLM. To install, place the diskette in the diskette drive and enter the following:

```
LOAD A:/INSTDSM
```

This will load the ADSM install program and display an initial panel where you can select the action to be performed. This screen is the same on all supported versions of NetWare:

```
        ADSM Install Options
      ========================
        Install ADSM
        Install SMS Modules
        Exit ADSM Install
```

If you select *Install ADSM*, the install program determines on which version of NetWare it is running and installs the required modules. It creates as a default a SYS:ADSM directory where all ADSM code is installed and copies the required SMS modules to the SYS:SYSTEM directory.

Note: An alternative directory for the ADSM code can be chosen to override the default SYS:ADSM directory.

You can also use the ADSM install program to install the NetWare SMS modules on a server by selecting *Install SMS Modules*:

```
        ADSM Install Options
      ========================
        Install ADSM
        Install SMS Modules
        Exit ADSM Install
```

This selection will install the SMDR and TSA NLMs in the SYS:SYSTEM directory. This installation must be done on the NetWare server where the NetWare ADSM client is installed. The installation can also optionally be done on other NetWare servers that the ADSM NetWare client will manage. The installation is self-configuring, in that ADSM determines on which version of NetWare it is running and selects the appropriate modules.

When the required modules have been installed, select *Exit ADSM Install* to exit. This is the end of the physical install process and the last time that physical access to the NetWare server is required.

3.1.1 Updating the Server AUTOEXEC.NCF File

After the ADSM code has been installed in the chosen directory and has replaced the SMS modules with the latest levels supplied with ADSM, you must update the NetWare server's AUTOEXEC.NCF file to reflect where ADSM has been installed.

Use the AUTOEXEC.NCF file to automatically load NLMs at server startup. By default, NetWare expects to find all of the NLMs in the SYS:SYSTEM directory. If ADSM is installed in the default SYS:ADSM directory, NetWare has to be told to search there for the DSMC.NLM. So you need to add a search path statement, such as:

```
SEARCH ADD SYS:ADSM
```

Note: DSMC is the name of NetWare ADSM client.

You can update the AUTOEXEC.NCF file in a number of ways:

1. Directly edit it from a requester with a text editor. AUTOEXEC.NCF is located in the SYS:SYSTEM directory. You can also edit it from the NetWare server using EDIT.NLM.
2. Load the INSTALL NLM, either at the NetWare server console or from a remote console session, and choose *System options*. From here you can edit the file.
3. Execute the SYSCON utility from a NetWare requester and invoke the *Supervisor options*. From here you can edit the file.

To perform any of these update options you must either be at the physical NetWare server console or be using the SUPERVISOR or another ID with equivalent security rights. Typical users cannot access the SYS:SYSTEM directory and cannot use RCONSOLE or the supervisor options of SYSCON.

Optionally, you can update the AUTOEXEC.NCF to automatically start ADSM and load the SMS modules. Place the following statement in the AUTOEXEC.NCF, after the SEARCH ADD statement:

```
LOAD DSMC
```

This will load the ADSM client at server startup and display the ADSM screen with a DSMC> prompt. An alternative is to add a statement to load the SMS modules to ensure that they are always loaded, which is particularly important if ADSM is run on another NetWare server. Loading DSMC only loads the SMS modules on the same physical server. To automatically load the SMS modules enter one of the following in the AUTOEXEC.NCF:

- LOAD TSA311 - for a NetWare 3.11 server
- LOAD TSA312 - for a NetWare 3.12 server
- LOAD TSA400 and LOAD TSANDS - for a NetWare 4.01 or 4.02 server.

Loading the TSA modules automatically loads the correct SMDR module for the NetWare server version being used.

Finally, when the AUTOEXEC.NCF file has been satisfactorily updated, it needs to be executed. It is normally executed only when the server is started. You can shut down the server by entering DOWN at the server console. After the server has shut down, you can enter the following two commands to restart it:

- EXIT followed by SERVER.

EXIT exits back to DOS, and the SERVER command restarts the NetWare server. When the server is restarted, the updated AUTOEXEC.NCF will be executed normally.

3.1.2 Updating the ADSM Options File

Having installed the ADSM client and updated the NetWare system, the next task is to configure the ADSM client. In common with other ADSM client platforms, the configuration data is in a file called DSM.OPT. On some ADSM client platforms there is some flexibility as to the name and location of this file.

With the NetWare ADSM client this is not possible. ADSM when installed provides a sample options file called DSM.SMP in the installed directory. This can be copied as DSM.OPT and used as a template. The options file must be called DSM.OPT and must be located in the same directory from which the DSMC NLM is loaded.

More than 40 different options can be specified in DSM.OPT, varying from the format of dates and numbers to connectivity settings between the ADSM client and server. The majority of the options have default settings that are adequate to start with.

Below we look at the options required to get the ADSM client working and those that are unique to NetWare or have specific meanings for NetWare. In subsequent sections of this chapter we look at the connectivity options that need to be added to the DSM.OPT file for the various connectivity methods.

The following options should be updated in the newly created DSM.OPT file:

NODENAME This option is required and identifies the ADSM client to the server. It can be up to 64 characters long. A sensible naming standard is to use the name of the NetWare server on which you are running the ADSM client for the node name, for example, ITSC-NW1.

This option specifies the node or client name that the ADSM administrator must register and assign a password to on the ADSM server. There is no default value for this option.

DOMAIN This option identifies to ADSM the NetWare volumes that are to be included for incremental backup processing. It can be used to add or exclude volumes from backup processing. For the ADSM NetWare client a fully qualified *Servername**Volume:* name can be used to identify volumes on multiple servers that the ADSM client can back up using the NetWare SMS API. For example:

```
DOMAIN    ITSC-NW1\SYS: ITSC-NW1\BINDERY
DOMAIN    ITSC-NW2\SYS: ITSC-NW2\BINDERY
DOMAIN    ITSC-NW3\SYS: ITSC-NW3\DIRECTORY
```

This example sets the domain to include volumes on three NetWare servers: ITSC-NW1, ITSC-NW2, and ITSC-NW3. The first two examples are NetWare 3.1x servers where the DOMAIN statements include NetWare volume names and the bindery. The last example is a NetWare version 4 server where the domain statement includes a NetWare volume and the directory, NDS.

Note: The correct syntax for DOMAIN includes a colon (:) at the end of the volume names, but not for the bindery or the directory statements.

Multiple domain entries can be put in the DSM.OPT file. The default for DOMAIN if not entered is all of the local volumes where the ADSM client is running. On a NetWare version 3.1x server this would be the SYS: and any other volumes, plus the bindery. On a NetWare version 4 server, only NetWare volumes are included. The directory must be added as a separate DOMAIN statement.

INCLUDE/EXCLUDE These options are used to include or exclude files from being backed up. They enable a greater level of granularity than using the DOMAIN option to include or exclude entire volumes. A fully qualified server and volume name can be used in these statements. For example:

```
EXCLUDE     ITSC-NW1\SYS:/.../*.NLM
EXCLUDE     ITSC-NW2\SYS:/.../*.NLM
EXCLUDE     ITSC-NW3\SYS:/.../*.NLM
EXCLUDE     ITSC-NW1\SYS:/ADSM/*
INCLUDE     ITSC-NW1\SYS:/ADSM/*.LOG        SPECIAL
INCLUDE     ITSC-NW1\SYS:/ADSM/*.OPT        SPECIAL
```

Again as with the DOMAIN statement, files on different NetWare servers can be identified. In this example all files that have a file extension of .NLM located on the SYS: volume on all three servers are excluded from being backed up. All files in the SYS:/ADSM directory on the ITSC-NW1 server are excluded, with the exception

of files with extensions of .OPT and .LOG. These files are included and will be backed up using a management class of SPECIAL.

The processing flow of INCLUDE/EXCLUDE statements is from the bottom up. Therefore, in this example, the two include statements will be effective because they are processed before the general exclude statement for their directory.

NWPWFILE This is an option unique to the ADSM NetWare client. When the ADSM NetWare client is loaded, it does not connect to either the ADSM server or the NetWare server itself. When a command such as INCREMENTAL is entered to perform an incremental backup, ADSM does two things. First, it connects with the ADSM server using the node name to identify itself. Second, it attaches to the NetWare server through the SMDR and TSA modules.

The first time an ADSM command is entered, the person who is using ADSM is prompted for a userid and password for the NetWare server being attached, as a security measure to ensure that only valid users attach to NetWare servers. The userid and password process is a NetWare security feature with which ADSM has to cooperate.

The NWPWFILE option allows the userid and password to be stored in an encrypted format in a file in the directory where the ADSM client is installed. Therefore after the first connection to a NetWare server, the user is no longer prompted for a NetWare server userid and password.

Instead the ADSM client obtains the userid and password from this file, de-encrypts them, and uses them to attach to the NetWare server. The advantage of the use of this option is that the scheduled ADSM operations can be performed without the need for a user to respond to a prompt when the operation starts.

The NWPWFILE option is either ON or OFF. If set to ON, the encrypted userid and password are stored in a file with an extension of .PWD. A password file is created for each attached server. ON is the default, and the recommended option.

NWUSER Associated with the NWPWFILE option is the NWUSER option. This is an alternative method of providing the userid and password required to attach to NetWare servers. The userid and password are entered in the DSM.OPT file as parameters to the NWUSER option. For example:

```
NWUSER   ITSC-NW2\TIM: PASSWORD
```

This statement is read from the DSM.OPT file when the ADSM client attaches to the NetWare server. It is not recommended that the NWUSER option be generally used. It leaves a userid and password stored as plain text in a file that could be read. For most installations this would be an unacceptable security exposure.

These are the minimum DSM.OPT configuration options that you need to consider. The remaining options affect how the client works, rather than being required to make it work. In the sections that follow we look at how to configure the various connectivity options and what options to add to the DSM.OPT file for each of them.

3.2 IPX/SPX Connectivity

ADSM release 2 introduced a number of new ADSM servers, clients, and available connectivity options. Among these was the option to use Novell's IPX/SPX protocols between certain ADSM clients and servers. The ADSM NetWare client supports this option for connection to ADSM/2 and ADSM/6000 servers only.

The IPX/SPX protocols are the basic protocols used in a NetWare environment for communications between NetWare servers and user workstations, and between NetWare servers. There is currently no support for IPX/SPX protocols to MVS or VM. This is not an ADSM restriction; rather, MVS and VM do not provide support for IPX/SPX.

3.2.1 System Requirements

The use of IPX/SPX for ADSM is fairly simple. The NetWare server that will have the ADSM client code installed on it will already support and use IPX/SPX protocols. The requirements for the two supported ADSM server platforms are slightly more complex. An ADSM/2 server requires the IPX/SPX protocols to be implemented. This support is provided by installing the NetWare Requester for OS/2 version 2.01 or later on the ADSM/2 server.

An ADSM/6000 server also requires the IPX/SPX protocols. There is no NetWare requester for AIX. This support is provided by installing NetWare for UNIX or AIX NetBIOS and IPX/SPX Support/6000 on the same RISC System/6000 that runs the ADSM/6000 server. This support is an implementation of a NetWare version 3.11 server running on AIX. This version of NetWare, running on AIX, is a separate product and is not supported as an ADSM NetWare client.

3.2.2 Configuration

Configuration of IPX/SPX connectivity is straightforward. In our example we use an ADSM/2 server. The first thing that has to be done is to install the NetWare requester for OS/2 on the ADSM/2 server. This provides the protocol support.

The NetWare requester is required on the ADSM/2 server to provide the protocol support only. There is no requirement that it be used for anything else. There is no need to log in to this NetWare on the ADSM/2 system. Figure 19 on page 49 illustrates the configuration described in this section. The ADSM server is an OS/2 system with the NetWare requester for OS/2 and the ADSM/2 server installed.

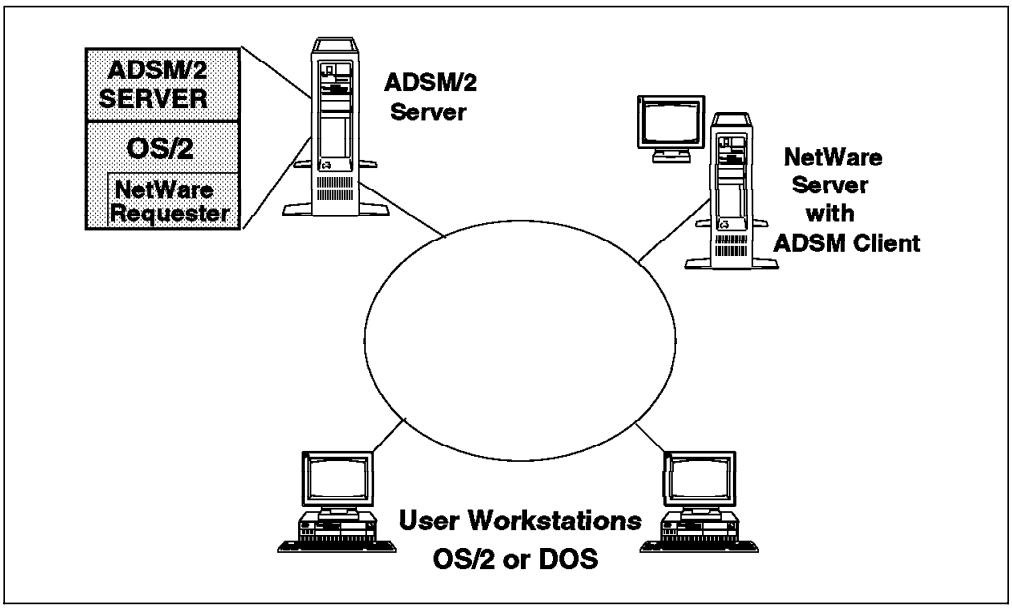


Figure 19. NetWare Server Overview

During the NetWare requester's installation process, a file called NET.CFG must be created with specific configuration data for the requester. In this file a link driver is specified defining frame protocols that will be used. Part of the link driver is the NODE ADDRESS that NetWare will use to communicate with the requester. For example:

```
...

LINK DRIVER LANSUP

    FRAME TOKEN-RING

    FRAME TOKEN-RING_SNAP

    NODE ADDRESS 400052047122

...
```

The node address is the token-ring address of the NetWare requester workstation and is used to override the burned in address on the token-ring adapter card. This node address is used later in the DSM.OPT file for the ADSM NetWare client.

When a NetWare server is first installed, a network address is defined for it. NetWare uses this address to identify a physical network segment where NetWare servers and requesters are located. The network address is defined in the server's AUTOEXEC.NCF file as part of the process of binding the IPX protocol to an adapter card. For example:

```
BIND IPX TO TOKEN NET=2
```

This statement binds the IPX protocol to an IBM token-ring adapter and defines a network address of 2. The NetWare network address is 8 characters long. When the network address is defined in the above manner, leading zeros can be omitted. In the above example the network address is actually 00000002.

The node address and the network address make up the IPX address that the ADSM NetWare client uses to communicate with an ADSM/2 server running the NetWare requester for OS/2. When concatenated together they make up a fully qualified name of a system:

NETWORK ADDRESS,NODE ADDRESS

The final component of IPX addressing is the IPX socket number. It is used to identify a process running on a system. ADSM uses an IPX socket on the ADSM/2 server to communicate with ADSM NetWare clients. As a default, ADSM uses a socket number of 8522 for the ADSM server. Numbers above 8000 are allocated by Novell. This number has been allocated for use by ADSM.

To configure the ADSM NetWare client the following, as a minimum, needs to be entered in the DSM.OPT file.

```
COMMMETHOD         IPXSPX

IPXSERVERADDRESS   00000002400052047122
```

The COMMMETHOD option defines IPX/SPX as the protocol. The IPXSERVERADDRESS is the network address followed by the node address of the ADSM/2 server running the NetWare requester.

There are two other related options, IPXSOCKET and IPXBUFFERSIZE. These have defaults that can be used to start. The socket number as previously stated is 8522, and the buffer size is 4KB. With the above two options in the DSM.OPT file, the ADSM NetWare client should be able to communicate with an ADSM/2 server.

3.2.3 Problem Determination

There are potentially a number of problems in running the ADSM NetWare client. First of all there might be difficulties getting an initial connection. The best way to prove that the basic connectivity is working is to log in to the NetWare server from the system on which the ADSM/2 server is installed, using the NetWare requester. If you can log in, the system should work. As long as the correct network address and node address are defined in the DSM.OPT file, the ADSM client should be able to communicate with an ADSM/2 server.

Once initial connectivity is established, problems might be experienced with reliability and performance. A number of PTFs are available from Novell for IPX/SPX. You should as a minimum obtain and apply a Novell PTF called STRTL2, which provides a number of fixes for IPX/SPX-related problems on NetWare 3.11, 3.12, 4.01, and 4.02 servers with IPX/SPX protocols.

3.2.4 Performance Considerations

The ADSM NetWare client option that you can specify for performance is the IPXBUFFERSIZE option of the DSM.OPT file. This option controls the amount of space to be used for IPX/SPX communications buffering. Communications performance can be improved with a larger buffer. The size of the buffer also

must be considered in regard to the overall memory requirements of the NetWare server.

IPX/SPX has a very small set of parameters to tailor its execution. You can control the IPX packet size at the NetWare requester by changing the IPX packet size limit in the NET.CFG file on the requester.

The IPX packet size is also controlled by the LAN driver used. This parameter defaults to either 4160, which is the optimum value for token-ring drivers, or the value specified by the LAN driver.

Other parameters control timeout and retry processing for IPX/SPX. In all cases it is best to use the default IPX/SPX values unless specific problems, such as running out of workstation memory, occur.

You can also control the transmission buffers used by the NetWare server for processing IPX/SPX packets. The size of the buffers must be large enough to accommodate the maximum packet size used by ADSM with IPX/SPX. A large buffer and packet size can speed communications but consumes more memory.

The maximum number of packet receive buffers can also be specified. NetWare dynamically allocates receive buffers based on need. The number of receive buffers controls the upper limit that the operating system can allocate. The MONITOR NLM can be used to monitor the *No ECB available count errors*. If the number of these errors is excessive, this parameter should be increased.

Maximum physical receive packet size and maximum packet receive buffers for a NetWare version 3.1x server are defined in the STARTUP.NCF file located on the server′s DOS partition. For a NetWare version 4 server they can be added to the AUTOEXEC.NCF file.

3.3 TCP/IP Connectivity

TCP/IP is the most common connectivity method available for ADSM. Just about every ADSM server and client combination can use it. The ADSM NetWare client supports TCP/IP connectivity from any supported NetWare server to any ADSM server.

In a similar way to IPX/SPX, TCP/IP consists of two protocols: IP and TCP. The Internet protocol (IP) is a connectionless protocol that provides the base network connectivity, but like IPX it does not guarantee delivery of IP packets. The transmission control program (TCP) provides the reliable transport mechanism, guaranteeing packet delivery. It runs on top of IP in a manner similar to SPX running on IPX.

3.3.1 System Requirements

The implementation of TCP/IP protocols on a NetWare server is simple. The TCP/IP protocols are provided as part of the NetWare operating system. No additional products are required; TCP/IP just needs to be configured. The requirements for the ADSM server vary and depend on the server platform:

- ADSM/6000 for AIX

  TCP/IP is included as part of the base AIX operating system.

- ADSM/2 for OS/2

  To use TCP/IP, TCP/IP version 1.2 or later for OS/2 is required.

- ADSM release 1 for MVS

  TCP/IP for MVS version 2.2 or later is required.

- ADSM release 1 for VM.

  TCP/IP for VM version 2.2 or later is required.

Note: There are alternative TCP/IP products for MVS and VM. They do not currently work with ADSM. ADSM on MVS and VM uses IUCV as the underlying communications method between TCP/IP and the ADSM server. Only TCP/IP products on MVS and VM that support IUCV will work with ADSM.

3.3.2 Configuration

Configuration of TCP/IP consists of two tasks: configuring NetWare to use TCP/IP and setting up the ADSM NetWare client to use TCP/IP.

NetWare versions 3.1x and 4 provide TCP/IP as part of their base operating system. They also support multiple protocols using the same physical network adapter. Therefore the token-ring adapter used for IPX/SPX traffic to and from user workstations can also be used for TCP/IP traffic for ADSM by binding another protocol to the card. Add the following entries in the server AUTOEXEC.NCF:

```
LOAD TOKEN FRAME=TOKEN-RING_SNAP NAME=TOKEN-TCP

LOAD TCPIP FORWARD=YES

BIND IP TO TOKEN-TCP ADDR=9.113.36.22 GATE=9.113.36.254
MASK=255.255.255.0
```

The first entry, LOAD TOKEN, loads the TOKEN token-ring adapter driver again but with a different frame type than is used for IPX/SPX traffic. This enables different data link control (DLC) frame types to be used with the same physical network adapter card. The parameters are the type of frame and a name for this logical adapter that has been loaded. TOKEN-RING_SNAP is the NetWare frame type for token-rings that support TCP/IP frames.

The second entry, LOAD TCPIP, loads the TCPIP modules on the NetWare server. The FORWARD=YES parameter allows the NetWare server to act, if required, as an IP router. Thus the NetWare server routes IP packets from one network to another.

The third entry binds the IP protocol to the logical network adapter previously created with the load token entry. In addition it sets up the following addressing-related parameters:

- ADDR=

  This is the IP address that you are creating for the NetWare server. It is required.

- GATE=

This is the IP address of the TCP/IP gateway that is used to gain access to other networks. If the ADSM server is on the same local network as the NetWare server, this address may not be required.

- MASK=

  This defines the subnet mask used to identify which portion of the IP addresses are networks and which are hosts. If subnetworking is being used, it is important that all systems use the same subnet mask.

The AUTOEXEC.NCF entries are sufficient to configure TCP/IP on a NetWare server. When the server is restarted, or the AUTOEXEC.NCF is executed, they will be activated.

The above example shows the configuration options in the server AUTOEXEC.NCF file. The options can also be executed at a NetWare server console. There are also UNLOAD and UNBIND commands to reverse the operations performed.

The second stage of configuring ADSM is to update the DSM.OPT file to reflect the fact that TCP/IP is being used. As a minimum you should add the following entries:

```
COMMMETHOD          TCPIP

TCPSERVERADDRESS    9.113.36.49

TCPPORT             1500
```

The first two entries identify the communication method and the IP address of the ADSM server. The third entry, TCPPORT, is used to identify the TCP/IP port that the ADSM server listens to for communications traffic from its clients. The TCP/IP port is something that is also configured on the ADSM server, and both must match. The default is 1500 for both the ADSM server and client configurations. However, port identification may be changed if the port numbers conflict with other applications.

The above configuration steps are sufficient to establish connectivity between an ADSM NetWare client and an ADSM server.

3.3.3 Problem Determination

Problem determination of TCP/IP connectivity failures can be difficult if complex network configurations are involved. Most TCP/IP implementations provide an application called Packet Inter Network Groper (PING). PING is a simple echo test where a packet is sent to a specified IP address and the response is displayed. PING is very useful for problem determination purposes if it is unclear whether a connection has been established.

The Novell implementation of TCP/IP on NetWare version 3.11 does not provide a PING application. This makes problem determination from NetWare rather more difficult. Novell does provide a PING.NLM with NetWare version 3.12 and version 4. PING can be used from any of the ADSM servers to contact the NetWare server. All the implementations of TCP/IP on the ADSM server platforms provide a PING application.

The NetWare TCP/IP implementation does provide a utility called TCPCON. This is a TCP/IP console where information concerning TCP/IP operations can be viewed. One of the useful functions provided is the capability to exchange simple network management protocol (SNMP) statistics between TCP/IP hosts. SNMP can be used as a simple alternative to PING.

TCPCON is an NLM that is used by entering LOAD TCPCON at the server console or a remote console. The TCP/IP console is then displayed. Under the *Available Actions* menu, a *Change Host* option is available. If this is selected, the TCP/IP address of a remote system can be entered. The upper half of the console will then display the TCP/IP statistics that are being exchanged with the remote system. If a connection cannot be made, a flashing HOST UNAVAILABLE message will be displayed.

Note: This facility only works if the remote system is running an SNMP agent. AIX provides this function. TCP/IP for OS/2 version 1 does not. TCP/IP for OS/2 version 2 does, as a configurable option. TCP/IP for MVS and VM provides this function through SNMP/NetView*.

3.3.4 Performance Considerations

It is beyond the scope of this document to provide comprehensive tuning guidance for TCP/IP. However, a number of basic steps can be taken on NetWare to maximize TCP/IP performance. In general the most significant TCP/IP performance gains can be achieved by modifying the following:

- Transmission block size
- Transmission buffers
- TCP/IP sliding window size
- TCP/IP buffers.

The transmission block size in TCP/IP is known as the maximum transmission unit (MTU) size. This is the size of the block of data that is sent between TCP/IP hosts. An acknowledgment is required after each MTU. The larger the MTU size, the fewer interruptions there are for acknowledgments, and the greater the overall data transfer rate.

There is a balance to be achieved in increasing the MTU size. MTUs are carried within data link control (DLC) frames across the physical network. If the MTU size is greater than the DLC frame size, segmentation occurs; that is, an MTU is broken up and sent in multiple DLC frames. This can result in loss of some of the benefits gained from using larger MTU sizes.

Maximum MTU sizes are defined on both NetWare and the target ADSM server system. The settings should match. A general recommendation is that the MTU size be set to 1500 for Ethernet LANs and 2000 for token-ring LANs. The packet size defined must include the header and trailer fields that encapsulate a 2000-byte packet.

Maximum MTU sizes

On a NetWare version 3.1x server the MTU setting must be added to the server's STARTUP.NCF file located on the server's DOS partition. For a NetWare version

4 server the MTU settings can be added to the AUTOEXEC.NCF file. To set the MTU for a NetWare server on a token-ring use the following parameters:

```
SET MINIMUM PACKET RECEIVE BUFFERS=200

SET MAXIMUM PHYSICAL RECEIVE PACKET SIZE=2040
```

The TCP/IP *sliding window* size also affects TCP/IP performance. TCP/IP is a connection-oriented protocol where every packet that is sent must be acknowledged before the next one is sent. The sliding window enables multiple packets to be sent without waiting for acknowledgments. Figure 20 Illustrates how a sliding window operates.

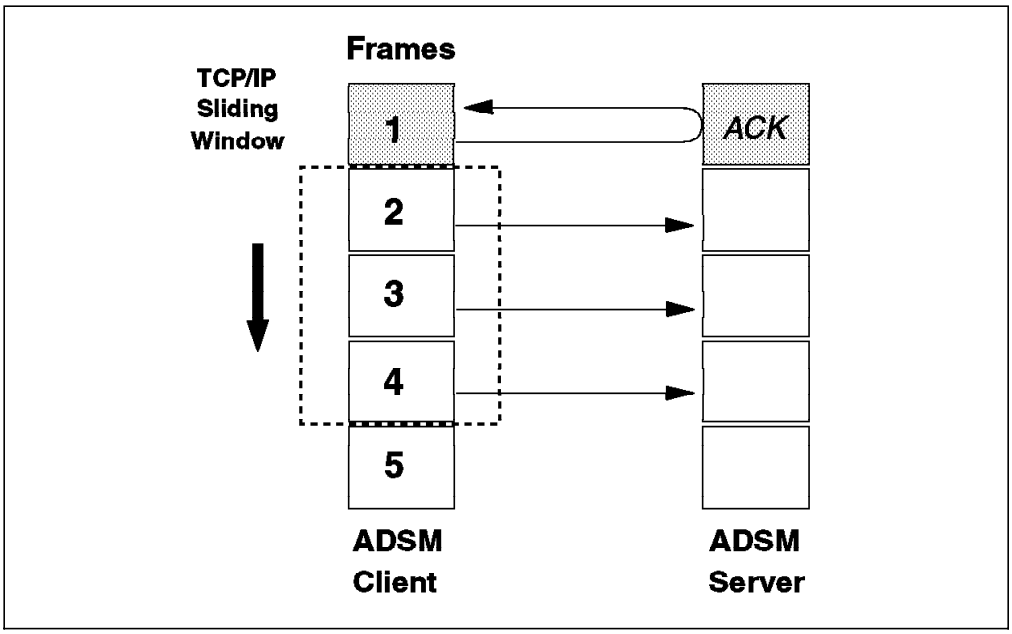


Figure 20. TCP/IP Sliding Window

The size of the sliding window is the amount of data that can be sent without waiting for an acknowledgment. When that amount of data has been sent, TCP/IP waits for the acknowledgments to start arriving. As they arrive, it slides the window down the frames waiting to be transmitted.

In Figure 20 the sliding window is large enough to contain three frames. The first three frames have been transmitted, and an acknowledgment has been received for the first frame. The window has now moved forward, and frame 4 has been sent. As the acknowledgments arrive, the window carries on moving down the frames, potentially sending multiple frames simultaneously.

In networks where long transmission delays occurr, a large sliding window can improve performance. With ADSM the sliding window size, in KB, is defined in the DSM.OPT file:

```
TCPWINDOWSIZE    24
```

The default is 16KB. Increasing the size to between 16 and 24KB is recommended.

The TCP/IP buffer size also affects performance. This parameter determines the size of the internal TCP/IP communications buffers to be used. The buffer size is defined in the DSM.OPT file:

```
TCPBUFFSIZE     16
```

The default is 8KB. Increasing the size to between 16 and 32KB can improve performance, although the buffer size has less effect on performance than the sliding window size.

Although both the sliding window size and buffer size can improve performance, they increase the use of memory on the NetWare server, which could cause other problems. A balance must be achieved between the use of NetWare server memory and communications performance.

The above general recommendations affect the TCP/IP performance of the NetWare server only. There are many other areas where tuning may be required: LAN gateways, routers, and the ADSM server. Recommendations for all of these areas are beyond the scope of this document.

3.4 APPC Connectivity

In this section we cover the APPC setup for ADSM NetWare clients, a subject that causes considerable confusion and misunderstanding. Part of the problem has to do with terminology, which we explain below.

At its simplest level, APPC is a conversation between two programs. These programs could be applications running on the same system or on remote systems connected by a network. In SNA terms the applications are defined as *logical units* (LUs). An LU is a port that can be addressed to access an application or device such as a screen or printer. APPC implements LU type 6.2.

There are two basic types of LUs: dependent and independent. *Dependent LUs* rely on a host system to establish and manage conversations between the LU and the application to which it is connecting. A system service control point (SSCP) manages these conversations. The SSCP is VTAM*.

Independent LUs are just that; they can talk to each other without an SSCP establishing and managing the conversation. An independent LU can activate a session with another independent LU directly. This process is called *binding*.

The only independent LUs are LU type 6.2. They can bind a session with another LU type 6.2 without assistance from VTAM. There are in fact, two types of LU type 6.2, dependent and independent. Dependent LU type 6.2 enable APPC conversations, but only under the control of VTAM.

ADSM implements independent LU type 6.2. Thus it can do any-to-any APPC connectivity between any of the ADSM server and client platforms that support APPC. In the sections that follow we concentrate on APPC connectivity to MVS.

To implement APPC connectivity for the ADSM NetWare client a number of pieces of information are required. Some will already be available, some will have to be defined. Figure 21 on page 57 illustrates some of the required parameters.

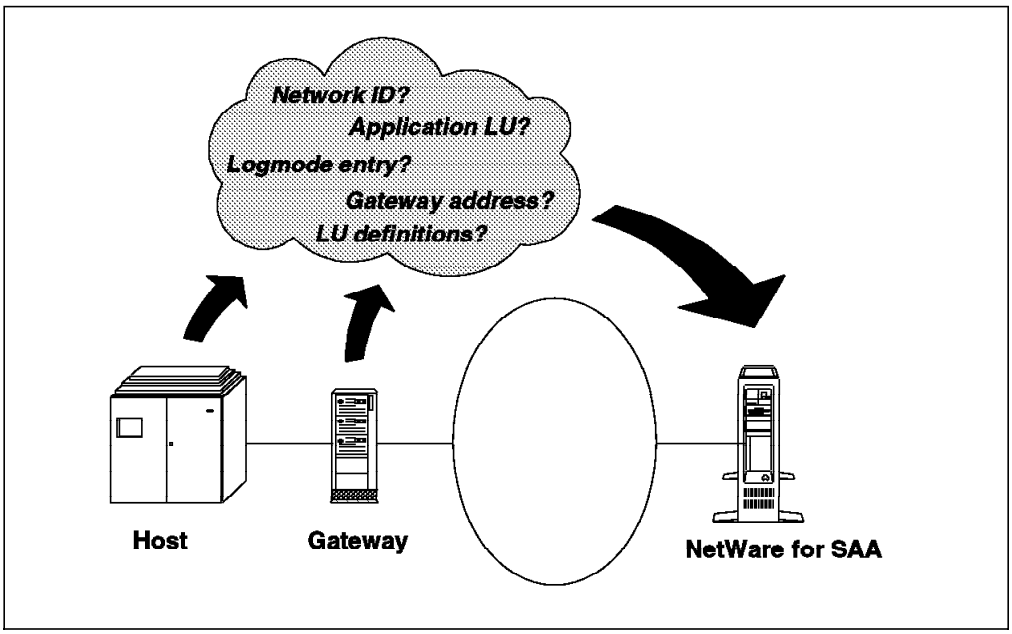


Figure 21. APPC Configuration Parameters

The following definitions are required:

- VTAM network ID

  This ID defines the VTAM network that will be required when NetWare for SAA is configured. The NETID= parameter in the VTAM ATSTRxx start list defines the network name.

- ADSM server VTAM application major node

  All resources that access VTAM must be defined. These resources include applications that communicate with other applications or terminals. Applications are defined to VTAM in a VTAM application major node definition.

  This definition defines the characteristics of the application, such as whether APPC is used, whether parallel sessions are enabled, and which logmode is to be used. Applications are defined by an LU name in the application major node definitions. The ADSM server will have an LU defined, which is the *partner LU* that will need to be defined in NetWare for SAA.

- Name of the VTAM logmode table and entry for ADSM

  Part of the application major node definition for ADSM is the logmode that will be used. A logmode is a set of parameters that govern the conversations between application LUs and other LUs. It contains parameters that determine the type of LU that can be used, network priorities, and maximum frame sizes.

  The logmode that the ADSM server uses must match the logmode used by the related LU, such as NetWare for SAA. This exchange of logmode information when establishing a session is part of the *bind*. If there are incompatibilities in the logmodes, the bind will fail. This is one of the most common problems encountered when setting up APPC.

  Logmodes are defined in tables that contain a number of entries. The ADSM server application major node defines a logmode table and the logmode entry within it to use. NetWare for SAA must specify the same logmode entry name as that used by the ADSM server.

- VTAM PU/LU definitions for NetWare for SAA machine

  The NetWare server running NetWare for SAA must be defined to VTAM. Typically, it will be defined as a *physical unit* (PU). The PU defines how VTAM accesses the system and controls the flow of data. Associated with the PU definitions are one or more LU definitions. These define the sessions that will be used with the PU.

  Multiple LUs can be defined for a PU. For ADSM the PU must have an independent LU type 6.2 defined. This independent LU is defined by adding a LOCADDR=0 statement to the LU. Adding this statement to an LU automatically makes the PU a type 2.1 PU. A PU type 2.1 supports dependent and independent LUs. A PU type 2.0 supports only dependent LUs and cannot be used for ADSM.

  The NetWare for SAA implementation of PU type 2.1 support only allows implementation of independent LUs. If NetWare for SAA is configured as a PU type 2.1, only independent LU type 6.2 sessions can be used on that PU. If dependent LUs for 3270 access are required, a separate PU type 2.0 must be defined.

- Address of host communications gateway

  The physical address of the communication gateway that will be used to access the VTAM network must be defined. This address is typically the token-ring address of a 3745, 3174, or other gateway.

A sample set of VTAM definitions can be found in Appendix A, “Sample NetWare APPC Configuration Definitions” on page 107. These definitions are used in the sections that follow.

3.4.1 System Requirements

The APPC connectivity requirements for ADSM are quite simple. An ADSM system uses independent LU type 6.2 for communication between ADSM servers and clients. How that level of support is provided is not so simple. There are three basic components that you need to consider when deciding how to implement APPC connectivity. These are:

1. Communications gateways
2. Host software
3. Workstation software.

The choice of connectivity protocol for ADSM is typically determined by the desire to utilize existing networks and investments in hardware and software. For that reason ADSM clients on NetWare servers that use APPC will typically be connected to ADSM MVS servers. Figure 22 on page 59 illustrates the basic connectivity options for ADSM NetWare clients connecting through APPC to an ADSM MVS server.

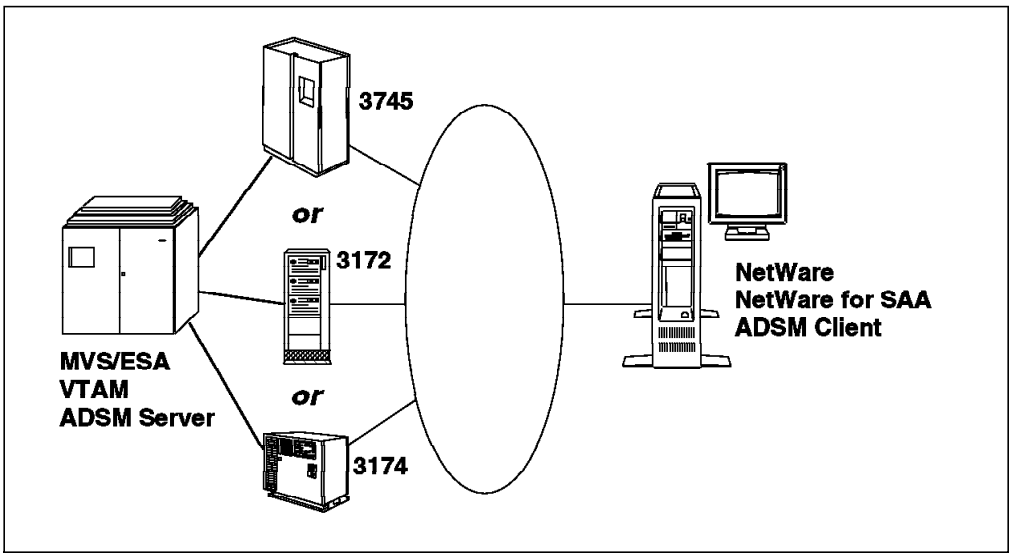


Figure 22. APPC Options for ADSM NetWare Client and MVS Server

In an ADSM environment there are many different ways of providing the required APPC connectivity. The primary requirement is for PU type 2.1 support in all components of the network. Without this support, APPC does not work. Independent LU type 6.2 can be defined only with PU type 2.1 support.

In an MVS environment PU type 2.1 support is provided as a combination of communications gateway software and VTAM levels. Let us look at the PU type 2.1 support prerequisites for the possible communication gateways in turn:

- 3745 communication controllers

  NCP version 5 release 2 and VTAM version 3 release 2 (VTAM/XA) or later are required.

- 3172 interconnect controllers

  Interconnect Controller Program version 2 and VTAM version 3 release 4 (VTAM/ESA) or later are required.

- 3174 workstation controllers

  3174 Licensed Internal Code, Configuration C release 1 (recommended) or 3174 Licensed Internal Code, Configuration B with RPQ #8Q0800, Type 2.1 Passthru Function, and VTAM version 3 release 4 (VTAM/ESA) or later are required.

On a NetWare server the NetWare for SAA product is required. The minimum level supported is version 1.2. However, a large amount of maintenance needs to be applied to that level. Thus, it is recommended that version 1.3B be used at a minimum. The level can be checked at the NetWare server console by loading the INSTALL utility. At the install screen, select *Product Options* and check the level of the COMMEXEC and NWSAA products. They should be version 1.3.42 or later.

NetWare for SAA and the ADSM NetWare client must be run on the same system. A remote gateway system cannot be used. ADSM and NetWare for SAA communicate with each other on the server bus as described in 2.1.1, “Server Architecture” on page 23.

3.4.2 Configuration

Configuring NetWare for SAA to support APPC consists of three steps:

1. Configuring the server AUTOEXEC.NCF file
2. Creating a *service profile*
3. Creating a *side information file*.

Each step is covered below. Detailed coverage is not given to installing the NetWare for SAA product. Only the configuration steps to enable APPC connectivity are covered.

3.4.2.1 Configuring the AUTOEXEC.NCF

As with other communications protocols, NetWare has to be configured to use APPC. Configuration is done in the NetWare server AUTOEXEC.NCF file. Before NetWare for SAA is installed, add the following entries to the AUTOEXEC.NCF file:

```
LOAD CLIB

LOAD SPXS

LOAD BTRIEVE -U=1 -P=4096 -F=20 -H=60 -L=20 -C
```

These entries may already be there for other purposes, but you should check to make sure that they are there. They enable the BTRIEVE database system that NetWare for SAA uses internally.

CLIB is the C interface library that is required with NetWare for SAA and its use of BTRIEVE. SPXS is also required. It is an NLM that provides support for communication streams through SPX. The parameters on the LOAD BTRIEVE command are those that are commonly used with BTRIEVE and do not need to be changed.

After NetWare for SAA has been installed, modify the AUTOEXEC.NCF for the SNA protocols. Here is an example for a server with a single token-ring card used for both LAN and SNA traffic:

```
LOAD TOKEN FRAME=TOKEN-RING SLOT=5 LS=24 SAPS=3 NODE=4011
NAME=TOKENR

BIND IPX TO TOKENR NET=2
```

These statements associate the token-ring card with the existing network, NET=2, used by IPX/SPX. The parameters on the LOAD TOKEN statement allow the token-ring card to by shared between IPX/SPX and APPC.

The key parameter additions for APPC support include the number of service access points (SAPs) and the number of link stations (LS). NetWare for SAA uses one SAP for each PU configured and one SAP per protocol bound to the card. In addition, NetWare for SAA uses eight link stations per SAP. With the number of SAPs equal to 3, we allow for two PUs for APPC as well as IPX/SPX and TCP/IP.

The final statement that you should add to the AUTOEXEC.NCF file is a statement to load the communications executive, COMMEXEC. COMMEXEC is one of the components that is installed with NetWare for SAA and provides the base communication services for NetWare for SAA. It is an NLM and can be automatically loaded by adding the following statement to the AUTOEXEC.NCF file:

```
LOAD COMMEXEC
```

The STARTUP.NCF file should be edited to enable additional buffers and set the maximum frame size to 4 KB:

```
SET MINIMUM PACKET RECEIVE BUFFERS=500

SET MAXIMUM PHYSICAL RECEIVE PACKET SIZE = 4200
```

Note: The above changes should have been made as part of of the NetWare for SAA installation. Appendix B, “Sample NetWare Server Configuration Files” on page 111 contains the STARTUP.NCF and AUTOEXEC.NCF files used during this configuration.

3.4.2.2 Creating a Service Profile

After NetWare for SAA has been successfully installed, you must define a service profile, which in SNA terms is the PU definition. NetWare for SAA allows a maximum of two service profiles to be loaded. The NetWare ADSM client requires a PU type 2.1 profile.

Appendix A, “Sample NetWare APPC Configuration Definitions” on page 107 contains sample VTAM definitions for PU type 2.1 nodes. The sample definitions for a 3745 switched major node are used in the setup example that follows. Throughout the example various parameters are highlighted and marked with references, **X**, so you can cross-reference them in the appendix.

From this point on, all configuration is performed from a NetWare requester workstation logged in to the NetWare server running NetWare for SAA. To create a service profile use the CSCON.EXE utility, which is located in the SYS:SYSTEM\CSCON directory.

To execute CSCON you must be the NetWare supervisor or have equivalent security rights. CSCON when executed prompts for this userid. After a valid userid has been entered, CSCON presents a NetWare menu with a prompt to select the communications services to be configured.

Select *NetWare for SAA*. At this point a list of existing service profiles is displayed. To create a new profile press the <INSERT> key. The screen will look like this when completed:

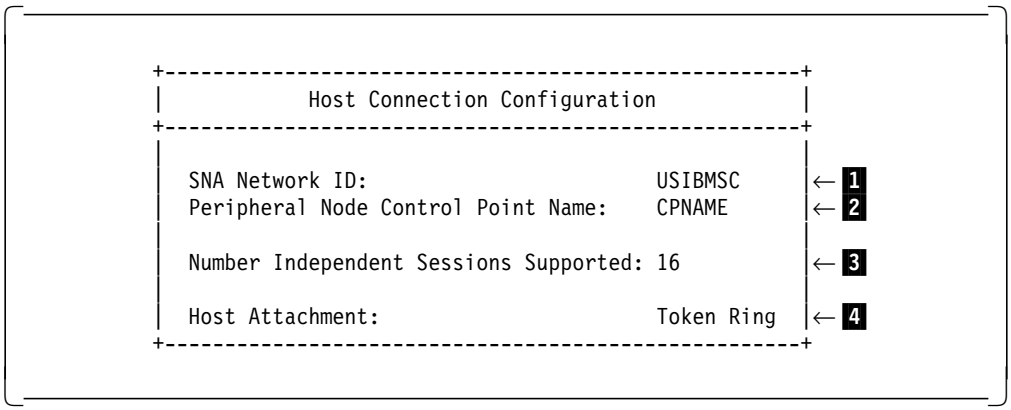
```
+----------------------------------------+
|           New Service Profile          |
+----------------------------------------+
| New Profile Name:            ADSM      |
| Node Type:                   PU 2.1    |
| Copy from Existing Profile:  No        |
+----------------------------------------+
```

Note: In all of the sample screens, CAPITAL letters are used. NetWare for SAA is case sensitive and can recognize trailing spaces as characters. Therefore it is strongly recommended that all entries in NetWare for SAA definition screens be in uppercase with no trailing spaces to avoid a lot of problems.

Enter a name for the service profile to be created. In the example above, the service profile name is ADSM. Move the cursor down to the *Node Type* field and press <ENTER>. A choice of profile types is displayed. Select PU type 2.1 and press <ESC> to exit this window.

You are asked to confirm that you want to create the service profile. Selecting YES returns you to the list of service profiles, including the profile just created.

Use the cursor down keys to select the new profile and press <ENTER>. A menu is then presented with a number of options. Choose *Configure Host Connection*. The screen looks like this when completed:



In the sample host connection configuration the following values have been entered:

1 SNA Network ID: *USIBMSC*

This value is the VTAM network ID for this workstation. It is defined in the VTAM ATSTRxx start list.

2 Peripheral Node Control Point Name

CPNAME is the default value entered by CSCON; it was generated by the system.

CPNAME is an alternative method of identifying a PU. A CPNAME can be defined in VTAM as part of the PU definitions for the NetWare for SAA server. In this example it has not been used.

If used this field must match the value used for the CPNAME in the PU definition of the ADSM client in VTAM. The CPNAME parameter must be used in the PU definition if IDBLK and IDNUM are not used. The CPNAME should have as its value the label name of the independent LU used by the ADSM client for APPC.

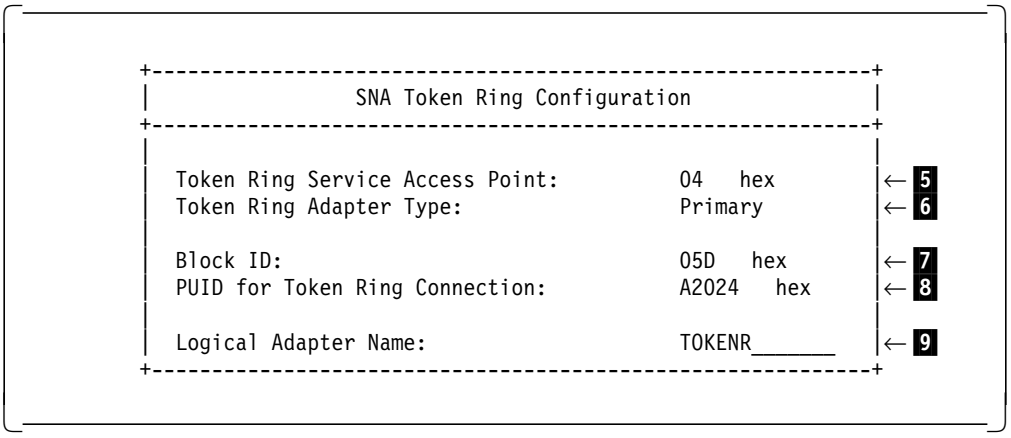
3 Number Independent Sessions Supported: *16*

This value is the number of LU type 6.2 sessions for this service profile. In this case we entered 16, which is the maximum for the NetWare for SAA license we were using, a 16-session license.

4 Host Attachment: *Token Ring*

This value is the type of connection to the host system. Pressing <SPACE> displays the choice of options available.

When the host attachment is selected, a second panel is displayed:



The following values have been entered for the sample SNA token-ring configuration:

5 Token Ring Service Access Point: *04*

This value was left to the default setting of 04 because in this example it was the only host connection. A setting of 04 must be coded if the destination is an MVS ADSM server with NCP and the PU is type 2.1. If a second service profile for a PU type 2.0 were to be defined, different and unique SAPs would have to be defined for its source and destination service access points.

6 Token Ring Adapter Type: *Primary*

This value was left to the default setting of primary. When you install and define the token-ring card, you select either primary or alternate for the card.

7 Block ID: *05D*

This value must match the IDBLK parameter for the NetWare for SAA PU definition, defined in VTAM.

8 PUID for Token Ring Connection: *A2024*

This value must match the IDNUM parameter for the NetWare for SAA PU definition, defined in VTAM.

Block ID and PUID for Token Ring Connection are the alternative to the CPNAME, **2**, method of identifying the PU definitions in VTAM that will be used. It is normal to use one or the other method but not both.

9 Logical Adapter Name: *TOKENR*

This value is the name of the token-ring adapter defined in the AUTOEXEC.NCF file.

When all fields are complete, press <ESC> to enter the values and save the changes. At this point the service profile has been defined and can now be loaded. At the NetWare server console or the remote console enter:

CSLOAD *SERVICE PROFILE NAME*

The console indicates that the profile has been loaded and activated:

```
:csload adsm
CE: Loading NWSAA profile <adsm>
NetWare for SAA
(c) Copyright 1990 - 1992.
Novell, Inc. All rights reserved
Version 1.30
SNA: NetWare for SAA Name is ADSM
APPC API for SAA: Ready for Transaction Programs with SNA agent 'ADSM'
```

At this stage the service profile is ready to be used. The next stage is to create a side information file.

3.4.2.3 Creating a Side Information File

NetWare for SAA provides an SAA common programming interface for communications (CPIC) utility to create *side information files*. This CPIC utility enables products such as ADSM that use a CPIC interface to work with NetWare for SAA. The side information file contains the configuration information, such as partner LU names and mode names, required to establish a session.

The NetWare for SAA utility to perform the side information file definition is called SIUTIL.EXE and is located in the SYS:SYSTEM\NWSAA\CPIC directory. As with CSCON, it is a utility that is run from a user's workstation logged in to the NetWare for SAA server. After you enter SIUTIL, the following NetWare menu is displayed:

```
          +--------------------------------------+
          |     SIUTIL.EXE Program Main Menu     |
          +--------------------------------------+
          |   Create New Side Information File   |
          |   Edit Existing Side Information File|
          +--------------------------------------+
               +------------------------------+
               | Create Side Information File |
               +==============================+
               |FILE NAME ADSMAPPC.CPI        |
               +==============================+
```

Note: Again, as with CSCON, it is strongly recommended that all entries in SIUTIL be in uppercase with no trailing spaces.

On the menu select *Create New Side Information File* to create a new file called ADSMAPPC.CPI. Side information files must have an extension of .CPI, but they can have any file name. After you have created the new side information file, the following menu is displayed:

```
+----------------------------------------+
| Side Information Records Command Menu |
+----------------------------------------+
|  Add A Side Information Record         |
|  Delete A Side Information Record      |
|  Display A Side Information Record     |
|  Modify A Side Information Record      |
+----------------------------------------+
```

A side information file can contain many side information records. These records define the configuration for the APPC sessions. The record name is referenced in the ADSM DSM.OPT file. To create a new side information record, select *Add A Side Information Record*, and the first of three screens is displayed:

```
+-----------------------------------------+
|        Local LU Configuration Data      |
+-----------------------------------------+
|  Side Information Record Name:  POKAPPC |← 10
|                                         |
|    LU Name:             SJA2024I        |← 11
|    PU Name:             SJA2024         |← 12
|    Detach PU Type:      1               |← 13
|    Security Type:       0               |
|    User ID:                             |
|    PassWord:                            |
|    LU Local Address:    0               |
|    LU Session Limit:    10              |← 14
|    Network Name:        USIBMSC         |← 15
|                                         |
|   Enter    F1 = Help    F7 = Cancel     |
|         Ctrl PgDn = PLU Record          |
|                                         |
+-----------------------------------------+
```

Note: In this and subsequent screens, only fields that were changed are referenced. All others have been left to their default values. In all of the screens, you can press <PF1> for an explanation of the fields.

The following values have been entered for this example:

10 Side Information Record Name: *POKAPPC*

This value is the ID that will be used in the ADSM DSM.OPT file. It can be any name that is required. However, a good recommendation is to keep the record names different from the file name to avoid confusion.

11 LU Name: *SJA2024I*

This value is the name of the independent LU defined for the NetWare for SAA server as part of the PU and LU definitions in VTAM. This LU name must exist for the PU that is referenced by either CPNAME or BLOCK ID/PUID in the service profile previously created with CSCON. It must be an independent LU, defined by a LOCADDR=0 statement in VTAM.

An independent LU can also be defined dynamically to VTAM by adding the DYNLU=YES statement to its PU definition. You can enter any LU name here if you use DYNLU=YES. If you use DYNLU=YES, the CPNAME value, **2**, used in creating a service profile must match the CPNAME value in the PU definition in VTAM.

12 PU Name: *SJA2024*

This value is the PU definition for the NetWare for SAA server. Again it must match the PU name referenced by either CPNAME, **2**, or BLOCK ID/PUID, **7** and **8**, in the service profile previously created with CSCON.

13 Detach PU Type: *1*

This value is the method used for detaching LUs. The value of 1 allows LUs to be detached only if there are no outstanding conversations.

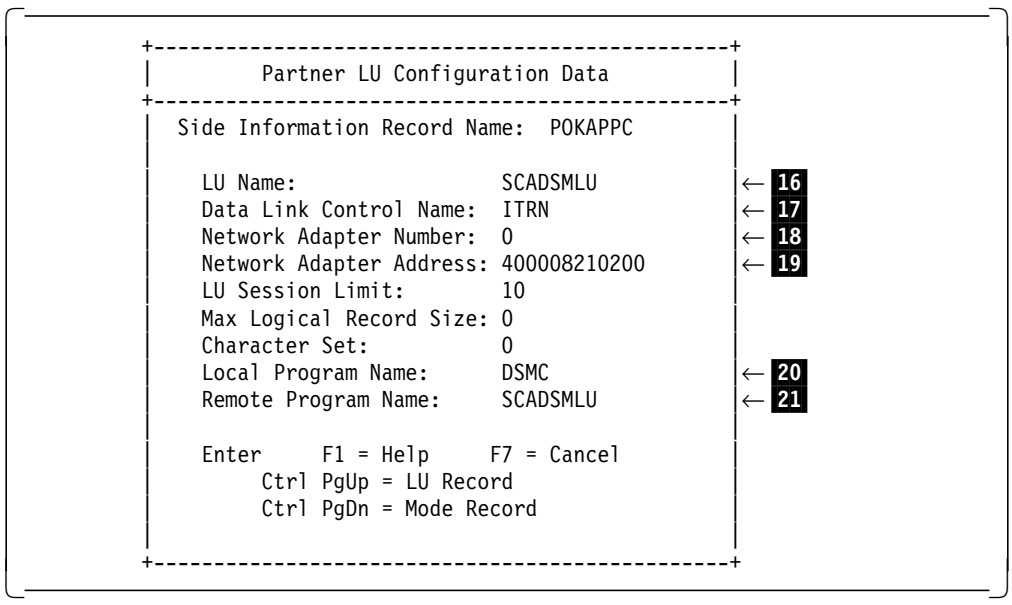
14 LU Session Limit: *10*

This value is the maximum number of sessions between the server and the partner LU. The value can be anything up to the limit allowed by the NetWare for SAA license.

15 Network Name: *USIBMSC*

This value is the VTAM network ID where the NetWare for SAA LU and PU, **11** and **12**, are defined when the ADSM client and server are in the same network. If this set of definitions is for a cross domain network environment, then this network name is the network ID of the ADSM server.

After entering these values page down to the next screen:



The following values have been entered:

16 LU Name: *SCADSMLU*

This value is the LU name of the ADSM server. It is defined in the VTAM application major node definition for ADSM.

17 Data Link Control Name: Set to *ITRN*

This value indicates a token-ring attachment.

18 Network Adapter Number: Set to *0*

This value is for a token-ring adapter defined as adapter 0.

19 Network Adapter Address: *400008210200*

This value is the token-ring address of the communications gateway to the host. In this case it is the token-ring address of the 3745 communications controller.

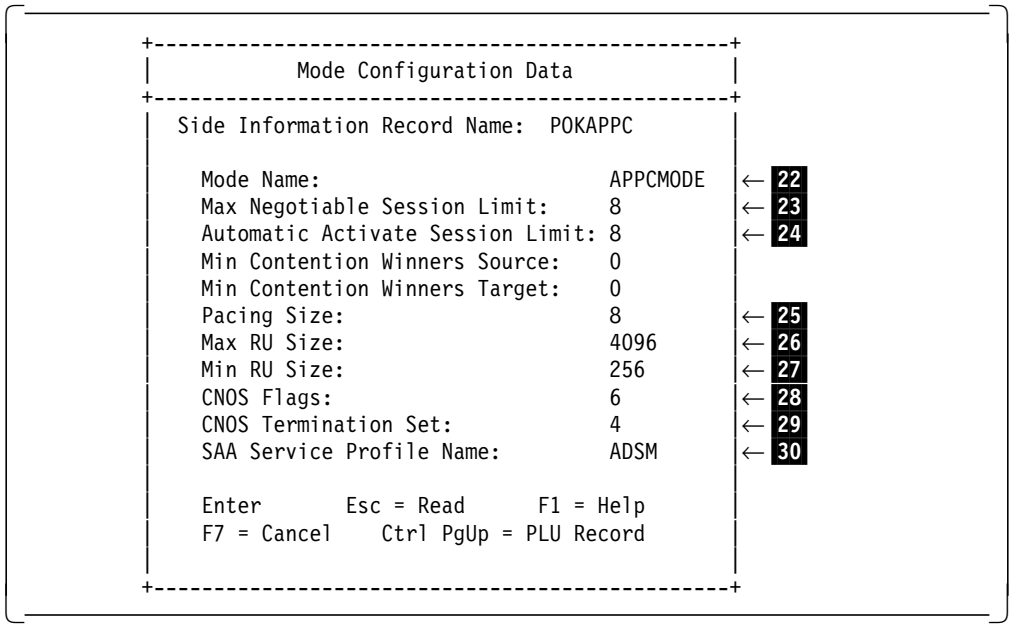
20 Local Program Name: *DSMC*

This value is the name of the application on the NetWare for SAA server that will use this LU type 6.2 session. The value is not used for anything, but a value must be entered. The name of the ADSM NLM was used.

21 Remote Program Name: *SCADSMLU*

This value is the name of the application that is associated with the remote LU. It is used only when VM is the ADSM server, and then it is the value of the APPCRESOURCE option in VM's DSMSERV OPT file. In this example it is not used, but a value must be entered. The ADSM server LU name was used.

When these values have been entered, page down to the final screen:



The following values have been entered:

22 Mode Name: *APPCMODE*

This value is the logmode entry that will be used during the bind process to establish the session. This name must match the logmode name that the ADSM server uses. It is defined with the DLOGMOD= parameter in the VTAM application's major node definition for the ADSM server.

23 and **24** Session Limits: *8*

These parameters control the number of sessions that can be negotiated during session initialization. The value of 8 was used for both; however, this value may need to be reviewed for individual implementations.

25 Pacing Size: *8*

The pacing size is the number of frames or response units (RUs) that can be received before a response is sent to the partner LU. Leaving the pacing size to the default of 0 allows VTAM to dynamically change the pacing depending on network activity. Setting it to a number overrides this dynamic

ability and fixes it at that value for the session. Leaving the pacing size at 0 is generally recommended.

26 and **27** RU Sizes: *4 KB* and *256* bytes

These values are the upper and lower limits for the size of the SNA RU between the local and remote LUs. The actual size used is negotiated during the bind process when a session is started. For ADSM the largest RU size that can be used is recommended. NetWare for SAA supports a maximum RU size of 4096 bytes.

28 and **29** CNOS Settings: *6* and *4*

These settings determine how sessions are negotiated, how session limits are set, and the rules for session termination. The values of 6 and 4 are good starting values.

30 SAA Service Profile Name: *ADSM*

This value is the name of the service profile defined previously with CSCON.

When all of these values have been entered, press <ESC> to process them. The <ESC> key then exits SIUTIL, and all the changes are saved. The side information created with SIUTIL now can be loaded to establish an LU type 6.2 session. To load the side information file enter the following command at the NetWare server console:

```
LOAD CPIC_SAA SYS:\SYSTEM\NWSAA\CPIC\ADSMAPPC.CPI
```

This command will load the previously created file and establish the LU type 6.2 sessions defined. If successful, the following screen appears on the NetWare server console:

```
:load cpic_saa sys:\system\nwsaa\cpic\adsmappc.cpi
Loading module CPIC_SAA.NLM
  NetWare for SAA CPI-C Module(v1.3.0)

Fri Jul  1 20:35:07 1994
802.2 LLC LINK ESTABLISHED.
ADAPTER NAME        : TOKENR
SOURCE ADDRESS      : 400000004011   SOURCE SAP      : 04
DESTINATION ADDRESS : 400008210200   DESTINATION SAP : 04
:
```

Note: The messages shown above are not displayed if you dynamically defined the LU by using DYNLU=YES on the PU definition in VTAM.

3.4.2.4 Updating the DSM.OPT File

The DSM.OPT file can be configured for APPC in one of two ways. You can specify either the symbolic destination ID or the full names of the VTAM network ID, partner LU name, and logmode name. It is recommended that the symbolic destination ID method, **10**, be used. The following entries would be added to the DSM.OPT file:

```
COMMMETHOD                SNALU6.2

SYMBOLICDESTINATION       POKAPPC    ← 10          .
```

Note: The value for the SYMBOLICDESTINATION is case sensitive. Take special care to ensure that this option is entered in a consistent manner in both NetWare for SAA and the DSM.OPT file.

This completes the setup of APPC connectivity for the NetWare ADSM client.

3.4.2.5 Unloading NetWare for SAA Profiles

You can deactivate NetWare for SAA side information profiles and service profiles by unloading them in the opposite way you used to load them. To deactivate the previously defined LU type 6.2 session enter the following at the NetWare server console:

```
:unload cpic_saa sys:\system\nwsaa\cpic\adsmappc.cpi
Module CPIC_SAA.NLM unloaded

Fri Jul  1 20:40:13 1994
802.2 LLC LINK DISCONNECTED.
ADAPTER NAME        : TOKENR
SOURCE ADDRESS      : 400000004011   SOURCE SAP      : 04
DESTINATION ADDRESS : 400008210200   DESTINATION SAP : 04
:
```

This deactivates the LU type 6.2 session between the local and remote LUs. Then you can unload the service profile by entering:

```
:csunload adsm
APPC API for SAA: Connection to SNA agent 'ADSM' terminated.
SNA: NetWare for SAA Agent, 'ADSM',  is unloaded.
:
```

3.4.3 Problem Determination

There are two scenarios where problem determination is likely to be required. The first and most common is where an APPC session cannot be established. The second is where an APPC session is established but does not perform adequately.

Determining why an APPC session does not work is not easy. It typically requires running a trace to ascertain the cause of the failure. The first things to check are the NetWare for SAA service profile and side information file definitions.

It is very easy to create errors in NetWare for SAA because the values for options are case sensitive. Ensure that all fields entered are in uppercase with no trailing spaces. If the definitions are correct and correlate with the VTAM definitions but do not work, you will have to run a trace.

Tracing also might be useful in determining the RU size that has been negotiated between NetWare for SAA and VTAM during the bind process. A small RU size is one of the basic causes of poor throughput.

Tracing is simple to perform but difficult to analyze. There are three ways to run a trace to analyze APPC problems with NetWare for SAA:

1. Run a VTAM trace from the host.
2. Run a PBTRACE at the NetWare for SAA server.
3. Run a CSSTATUS trace at the NetWare for SAA server.

The first option, running a VTAM trace from the host is not covered here because it is beyond the scope of this book. The second and third options are discussed below.

In general, problems where a session cannot be started will present sense information. These sense codes, created by VTAM, provide valuable information as to the cause of the problem. There is a facility in NetView where a VTAM sense code can be entered to NetView and then NetView provides a description of the problem. A VTAM or networking specialist can extract the sense codes from a trace.

3.4.3.1 NetWare PBTRACE

NetWare for SAA provides a PBTRACE NLM, a tool for tracing APPC sessions on NetWare for SAA. It is loaded as an NLM at the server console. The syntax of the command to load it is:

LOAD PBTRACE *SERVICE PROFILE NAME*

The service profile that is going to be traced has to be loaded before the trace is started. The following screen illustrates the sequence of actions at a server console:

```
:csload adsm
CE: Loading NWSAA profile <adsm>

NetWare for SAA
(c) Copyright 1990 - 1992.
Novell, Inc. All rights reserved
Version 1.30
SNA: NetWare for SAA Name is ADSM
APPC API for SAA: Ready for Transaction Programs with SNA agent 'ADSM'
:load pbtrace adsm
Loading module PBTRACE.NLM
  APPC API TRACE Driver (v1.3.01)
  Version 0.00    June 7, 1993
```

The LOAD CPIC_SAA command can now be used to load the failing side information file. PBTRACE creates its trace output in a file called OUTPUT.PC in the SYS:SYSTEM directory. This file is formatted and shows the APPC API calls made. An example of the output is shown below:

```
API req            00A2B33C     11:09:32.00
     00000000 00000000 00000000 20000000        <.............>
     CC39BB00 00000000 00000000 E4E2C9C2        <...........USIB>
     D4E2C340 E2D1C1F2 F0F2F440 00000000        <MSC SJA2024 ....>
     00000000 FFFFFFFF 00000000 00000000        <.............>
     00000000 00000000 00000000 00000000        <.............>
```

```
     00000000 00000000 00000000 00000000        <.............>

API ret            00A2B33C     11:09:32.00
     00000000 00000000 3CB3A200 20000000        <.........s.....>
     6CB2A200 00000000 04030610 E4E2C9C2        <..s.........USIB>
     D4E2C340 E2D1C1F2 F0F2F440 00000000        <MSC SJA2024 ....>
     00000000 FFFFFFFF 00000000 00000000        <.............>
     00000000 00000000 00000000 00000000        <.............>
     00000000 00000000 00000000 00000000        <.............>
```

Appendix C, “Sample NetWare PBTRACE” on page 115 contains a sample trace listing from a PTRACE. It is for a session that was successfully started using the configuration described in 3.4.2.2, “Creating a Service Profile” on page 61 and 3.4.2.3, “Creating a Side Information File” on page 64.

Although PBTRACE provides a lot of information, it is not easy to interpret. If you use it, a VTAM specialist will be required to interpret the results.

3.4.3.2 NetWare CSSTATUS

CSSTATUS is a NetWare for SAA utility that can be used to monitor the activity of NetWare for SAA sessions. It also provides the ability to trace sessions. CSSTATUS is invoked by entering CSSTATUS at the server console. It displays a NetWare menu showing the options available:

```
+-----------------------------+
|      CSStatus Options       |
+-----------------------------+
|  Service Options            |
|  NetView Management Options |
|  Log and Trace Options      |
|  Security Options           |
|  Exit CSStatus              |
+-----------------------------+
```

If *Service Options* is selected, a second menu is displayed where a trace can be started and stopped. Selecting *Start Trace* prompts you for a file name where the trace output is created. The LOAD CPIC_SAA command can then be used to load a side information file. To stop tracing, select *Stop Trace*:

```
+-----------------------------+
|      Service Function       |
+-----------------------------+
|  Display Status for Service |
|  Start Trace                |
|  Stop Trace                 |
+-----------------------------+
```

The CSSTATUS trace file that is created is not formatted. A separate utility can be used to format traces created with CSSTATUS. This utility is called TRFORMAT.EXE and is located in the SYS:SYSTEM\NWSAA\TRACE directory.

The TRFORMAT.EXE utility prompts you for the name of the trace file created by CSSTATUS and a name for the output file. This trace is formatted in a more understandable way than PBTRACE. An example of the format is:

```
<=S==   TIME:00:00:01.05    #:005e  TH: 2F0001040004 RH: 6B8000
 SEG=ONLY   EFI=ON  DAF:01  OAF:04  SEQ:0004
 RU=REQ,DR  RC=SC,BIND  CHAIN=ONLY
 RU: 31001307 B0B050B1 01008585 80010602 00000000 00000000 23000010 E4E2C9C2
     D4E2C34B E2D1C1F2 F0F2F4C9 27000902 E2D5C1E2 E5C3D4C7 090300B7 985B1AFA
     947C1104 E4E2C9C2 D4E2C34B E2D1C1F2 F0F2F4C9 0008E2C3 C1C4E2D4 D3E4

==R=>   TIME:00:00:02.10    #:0046  TH: 2F0004010004 RH: EB8000
 SEG=ONLY   EFI=ON  DAF:04  OAF:01  SEQ:0004
 RU=RSP+ RC=SC,BIND  CHAIN=ONLY
 RU: 31001307 B0B050B1 01028585 82010602 00000000 00000012 23000000 27000902
     E2D5C1E2 E5C3D4C7 0903F0B7 985B1AFA 947C1105 E4E2C9C2 D4E2C34B E2C3C1C4
     E2D4D3E4 0000
```

Appendix D, “Sample NetWare CSSTATUS Trace” on page 119 contains a sample listing from a CSSTATUS trace for a session that was successfully started using the configuration described in 3.4.2.2, “Creating a Service Profile” on page 61 and 3.4.2.3, “Creating a Side Information File” on page 64.

CSSTATUS trace provides a lot of information but is not easy to understand. If you use it, a VTAM specialist will be required to interpret the results.

3.4.4 Performance Considerations

The CPICBUFFERSIZE option in the DSM.OPT file sets the size of the APPC buffers used in the ADSM NetWare client. You should set the size to the highest value possible within the contraints of your ADSM client system′s memory.

Tuning APPC is a complex task. Many parameters in VTAM, NCP, and NetWare for SAA affect APPC performance. However, the following basic parameters typically make the most dramatic improvements:

- Number of DLC and APPC buffers
- DLC frame size
- RU size
- RU pacing.

The number of buffers available should be considered with ADSM. If an adequate number of buffers are not provided, communications will slow down. As the NetWare server is installed the number of DLC buffers is set. This value may reflect a normal interactive communications workload.

ADSM backup, recovery, archive, and retrieve are batch applications that perform better with a larger number of buffers, each of a larger size, than interactive applications. An example of how to increase the number of DLC buffers (packet receive buffers) is shown in 3.4.2.1, “Configuring the AUTOEXEC.NCF” on page 60. .

The token-ring LAN architecture allows token-ring adapters a maximum of 10 milliseconds to capture the token and transmit a token-ring frame behind the captured token. Therefore a workstation on a 4MB token-ring LAN can transmit a frame of up to 5,000 bytes. On a 16MB token-ring LAN the maximum DLC frame size is 20,000 bytes.

A larger DLC frame size produces a higher utilization of the LAN. The greater the number of bytes that can be transmitted in one packet, the less time the sending machine has to contend for bandwidth on the LAN. Also, RU segmentation is avoided with a larger DLC frame size.

A DLC frame size of 4KB is a good starting point for ADSM. The MAXIMUM PHYSICAL RECEIVE PACKET SIZE must be set to a large enough value to handle the DLC frame size. An example that allows a 4KB DLC frame size is shown in 3.4.2.1, “Configuring the AUTOEXEC.NCF” on page 60.

The APPC buffer size also must be considered. The CPICBUFFERSIZE option of the DSM.OPT file sets the size of this buffer. The larger this buffer is, the better communications performance is in general. If the NetWare server is short on memory, you should continue to use the default of 15KB. If the machine has ample memory, increase this option to the maximum of 31KB.

For bulk data transfer products like ADSM, a large RU size is desirable for performance. The RU size that can be used is negotiated as part of the bind process at the beginning of each session. The RU size used is the minimum value allowed by each node in the transmission path.

Increasing the RU size that a machine uses increases the amount of memory needed. A good starting point for ADSM is a maximum RU size of 4096 bytes and a minimum RU size of 256 bytes.

The RU size for the NetWare ADSM client is defined within NetWare for SAA in the side information file, **26** and **27**, as discussed in 3.4.2.3, “Creating a Side Information File” on page 64. The RU size should be coordinated between the SNA nodes between the ADSM client and server. For an MVS ADSM server, the RU size is set in the VTAM logmode entry.

RU pacing controls the number of RUs a machine can send before stopping data transmission and waiting to receive an acknowledgment. If a small window is specified, data transfer may intermittently stop in order to wait for acknowledgments.

Efficient memory usage is most important in small machines. To balance better performance with efficient memory usage, a maximum pacing window may not always be desirable. RU pacing should be matched to the power of the ADSM client workstation. A value of 8 is recommended for an 80386-based machine and 63 for an 80486-based machine.

RU pacing is set in the side information file, **25**, for NetWare for SAA. It is also defined with the APPC parameters for the ADSM server.

APPC also needs to be tuned in the ADSM server. In some case parameters such as the RU size should be coordinated between the ADSM client, the ADSM server, and any intermediate nodes for optimal performance.

A detailed discussion of APPC at the ADSM server is beyond the scope of this book. The type of parameters to consider when MVS is the ADSM server include the following VTAM parameters:

- IOBUF - the amount of buffering
- MAXDATA - maximum number of bytes VTAM receives in a request
- MAXOUT - maximum number of segments VTAM sends before a response
- PACING and VPACING - RU pacing
- RUSIZES - RU maximum and minimum sizes.

The key parameters within NCP include:

- MAXBUFRU - number of RU buffers
- MAXDATA - maximum number of bytes NCP receives in a request
- RCVBUFC - DLC frame size
- T2TIMER - DLC pacing
- UNITSZ - RU buffer size.

3.5 Starting the NetWare ADSM Client

In this section we explain how to start the NetWare ADSM client for the first time. At this stage the ADSM client has been installed, the required connectivity to the ADSM server has been set up, and the ADSM DSM.OPT file has been customized with the basic options required to start a session.

To start the ADSM client the DSM.OPT file should contain as a minimum the following options:

```
NODENAME                ITSC-NW3

DOMAIN                  ALL-LOCAL

COMMMETHOD              IPXSPX

IPXSERVERADDRESS        00000002400052047122

     or

COMMMETHOD              TCPIP

TCPSERVERADDRESS        9.113.36.49

TCPPORT                 1500

     or

COMMMETHOD              SNALU6.2

SYMBOLICDESTINATION     POKAPPC
```

Only one COMMMETHOD is allowed for the DSM.OPT file. This example shows the three communications protocols discussed in this chapter.

Note: It is recommended that all values entered for options be in uppercase. This is particularly important when using NetWare for SAA for APPC connectivity.

The NetWare ADSM client can now be started at the server console (or a remote console session) by entering the following command:

```
LOAD DSMC
```

The ADSM client NLM will load and display the ADSM console session. The NetWare ADSM client can be run only from a console or remote console session. The various console sessions can be viewed by pressing <CTRL> and <ESC>, then selecting the required screen.

Once the ADSM client has been loaded, it displays an ADSM prompt, DSMC>, and waits for a command to be entered. At this point no connection to the ADSM server has been established. If a command such as INCREMENTAL (to perform an incremental backup) is entered, a session with the server will be started:

```
ADSTAR Distributed Storage Manager
Command Line Backup Client Interface - Version 1, Release 2, Level 0.1
(C) Copyright IBM Corporation, 1990, 1994, All Rights Reserved.
dsmc> incremental
Please enter password for node "NW3": ******
```

The ADSM client establishes a session with the ADSM server and prompts for the client password. When the password has been validated, the ADSM client connects to the NetWare server by invoking the NetWare SMS API and connecting to the NetWare server's TSA. At this point the ADSM user is prompted for a NetWare userid on the target NetWare server:
.

```
ADSTAR Distributed Storage Manager
Command Line Backup Client Interface - Version 1, Release 2, Level 0.1
(C) Copyright IBM Corporation, 1990, 1994, All Rights Reserved.

dsmc> incremental
Please enter password for node "NW3": ******

Session established with server ADSM: MVS
  Server Version 1, Release 1, Level 0.4
  Server date/time: 07/01/1994 22:19:14   Last access: 07/01/1994 22:17:40


Connecting to a NetWare 3.11 File System (ITSC-NW3).


Please enter NetWare user for "ITSC-NW3": tim

Please enter the password on "ITSC-NW3" for NetWare user "TIM":
```

The userid and password are stored in an encrypted format in a file located in the ADSM directory if the NWPWFILE option is set to ON in the DSM.OPT. If this option is used, subsequent ADSM functions that require access to the NetWare

server will obtain the userid and password for this file rather than prompting for a NetWare userid every time.

The NetWare userid does not have to be an ID with supervisor equivalent authority. However, it should have at least read access to all files and directories that are going to be backed up. If this is not the case, ADSM failures will occur when an attempt is made to back up or archive files.

The NetWare ADSM client has now connected to the ADSM server and has started to perform its first incremental backup. To stop the ADSM client enter, QUIT at the DSMC> prompt or UNLOAD DSMC at the NetWare server console.

Chapter 4. Using the ADSM NetWare Client

In this chapter we look at how the NetWare ADSM client can be used on a NetWare server. We cover:

- Backing up and restoring data
- Archiving and retrieving data
- Running scheduled operations.

The NetWare ADSM client is started from the NetWare server console or a NetWare requester workstation running a remote console session. It is started by entering the following command:

```
LOAD DSMC
```

This command loads the NetWare ADSM client NLM (DSMC.NLM) and displays the DSMC> prompt on the ADSM screen. At this point, ADSM commands can be entered.

4.1 Backing Up and Restoring Data

Backup and restore are the ADSM functions that you will use most. In this section we briefly look at how these functions are performed with the NetWare ADSM client. We consider the different types of backup available, the different types of data on a NetWare server, and how to back up that data with ADSM.

4.1.1 Incremental Backup

An incremental backup is a backup of selected files and all directories. The decision to back up a file is made on the basis of whether the file has changed since the last time it was backed up. The file size, last modification date and time, and trustee rights are used to determine whether the file has changed.

The file only is backed up if the volume on which it resides has been identified for incremental backup processing and the file has not been excluded. ADSM also allows a minimum number of days to be defined between backups even if the file has changed. In this case the incremental backup will not occur until the minimum number of days has elapsed since the last backup was taken.

To perform an incremental backup you enter the following ADSM command at the DSMC> prompt:

```
INCREMENTAL
```

If incremental backup is the first function to be performed, the ADSM server will prompt for the ADSM password for this NetWare client. The NetWare ADSM client will then connect to the NetWare file system. Below is an extract of what will be seen:

```
ADSTAR Distributed Storage Manager
Command Line Backup Client Interface - Version 1, Release 2, Level 0.1
(C) Copyright IBM Corporation, 1990, 1994, All Rights Reserved.

dsmc> incremental
Please enter password for node "NW1": ***

Session established with server ITSC: OS/2
  Server Version 1, Release 2, Level 0.0
  Server date/time: 07/06/1994 11:29:58   Last access: 07/06/1994 11:29:00

Connecting to a NetWare 3.11 File System (ITSC-NW1).

Connected to ITSC-NW1.

Incremental backup of volume 'ITSC-NW1\SYS:'
ANS4102I ***** Processed       500 files *****
Successful incremental backup of 'ITSC-NW1\SYS:'

Incremental backup of volume 'ITSC-NW1\BINDERY:'
Normal File-->       2,048,000 ITSC-NW1\BINDERY:/BINDERY . Sent
Successful incremental backup of 'ITSC-NW1\BINDERY'
```

An incremental backup performs a number of functions:

1. It backs up the NetWare volumes specified in the DOMAIN statement in the DSM.OPT file.
2. For volumes being backed up, it backs up only those files that have either been included in or not specifically excluded from the DSM.OPT file's INCLUDE/EXCLUDE list.
3. For volumes being backed up, it backs up all directory entries even if the files within them have been excluded.

The NetWare ADSM client backs up the *trustee rights* and *attributes* for files and directories during every incremental backup. Thus, incremental backups run on a regular basis provide a high level of data protection.

4.1.2 Selective Backup

A selective backup can be used to back up individual files or groups of files whether they have changed or not. A selective backup is invoked using the SELECTIVE command followed by a file specification. Here is an example of a single file backup:

```
dsmc> selective sys:adsm\dsm.opt
Selective Backup function invoked.

Normal File-->          1,181 ITSC-NW1\SYS:/ADSM/DSM.OPT . Sent
Selective Backup processing of 'sys:adsm\dsm.opt' finished with no failures.

Total number of objects inspected:        1
Total number of objects backed up:        1
Total number of objects updated:          0
Total number of objects rebound:          0
Total number of objects deleted:          0
Total number of objects failed:           0
Total number of bytes transferred:    1,344
Elapsed processing time:            0:00:01
```

As with incremental backup, a selective backup backs up trustee rights and attributes along with the file. Selective backups do not override the INCLUDE/EXCLUDE list in the DSM.OPT file. If an attempt is made to back up an excluded file, the following occurs:

```
dsmc> selective sys:adsm\dsm.smp
Selective Backup function invoked.

ANS4228E Send of object 'ITSC-NW1\SYS:/ADSM/DSM.SMP' failed
ANS4119E File excluded from backup by Include/Exclude list
Selective Backup processing of 'sys:adsm\dsm.smp' finished with failures.
 .
 .
```

To override the INCLUDE/EXCLUDE list enter the following after the ADSM client is loaded:

```
LOAD DSMC  -INCLUDE="SYS:\ADSM\*     SPECIAL"
```

In this case the management class, SPECIAL, is assigned to an entire group of files.

You can also override the INCLUDE/EXCLUDE list after ADSM has been loaded by specifying the -INCLUDE option when you issue a command. If you override the INCLUDE/EXCLUDE list in this manner, you can back up files that would otherwise be excluded.

Exercise extreme care when overriding the INCLUDE/EXCLUDE list with the -INCLUDE option. This change to the INCLUDE/EXCLUDE list is valid only for the duration of a session. Once the client has been unloaded the change will be lost. The next time the client is loaded it uses the INCLUDE/EXCLUDE list in the DSM.OPT file.

4.1.3 Restoring Backup Data

You can restore backup copies of files and directories by using the RESTORE command. You can use this command to restore single files, multiple files, complete directories, or entire NetWare volumes. To restore a single file enter the following command:

```
dsmc> restore sys:\tim\itsc.scr
Restore function invoked.

Please enter password for node "NW1": ***

Session established with server ITSC: OS/2
  Server Version 1, Release 2, Level 0.0
  Server date/time: 07/06/1994 15:59:26   Last access: 07/06/1994 15:24:00

Connecting to a NetWare 3.11 File System (ITSC-NW1).

Connected to ITSC-NW1.

File ITSC-NW1\SYS:/TIM/ITSC.SCR already exists, do you want to replace it? (Yes/
No) yes
Restoring          189 ITSC-NW1\SYS:/TIM/ITSC.SCR . Done

Restore processing finished.
```

By default ADSM prompts you during a restore if the file already exists. You can choose to continue or bypass the restore.

To restore an entire directory and all of the files it references enter the following:

```
dsmc> restore sys:\tim\*.*
Restore function invoked.

Restoring            0 ITSC-NW1\SYS:/TIM . Done
Restoring          189 ITSC-NW1\SYS:/TIM/ITSC.SCR . Done
Restoring          475 ITSC-NW1\SYS:/TIM/ITSC2.SCR . Done
Restoring          189 ITSC-NW1\SYS:/TIM/TEST.SCR . Done

Restore processing finished.
```

This restore process re-creates the directory and all of its files. All trustee rights and attributes for the directory and files are also restored. The above command by default restores only one level of the directory structure. Subdirectories below \TIM would not be restored.

The ADSM option, SUBDIR, can be used to process subdirectories. You can specify SUBDIR in two ways:

- In the DSM.OPT file with the SUBDIR YES option added
- As an option specified with a command.

For the example above, if you enter the following command, all subdirectories and files would be included in the restore:

```
RESTORE  -SUBDIR=YES  SYS:\TIM\*.*
```

Taken a step further, you can use the SUBDIR option to restore an entire NetWare volume. For example, to restore the NetWare volume called APPL:, enter:

```
RESTORE  -SUBDIR=YES  APPL:\*.*
```

This command restores everything that has been backed up for the APPL: volume. You are prompted to confirm the replacement of every file that already

exists on the NetWare file system. To override this prompting for existing files, use the REPLACE option:

```
RESTORE  -SUBDIR=YES  -REPLACE=YES  APPL\*.*
```

This option restores everything that has been backed up for the volume, overwriting any files that still exist on the volume. The REPLACE option is very powerful and should be used with care. It can be very useful if you have to recover an entire volume.

4.1.3.1 Restoring Data to an Alternative Directory Structure

The restore examples discussed so far involve restoring various files, directories, or volumes to their original location, which is how ADSM restores data by default. However, ADSM also can restore data to a different directory or subdirectory. To restore the SYS:\TIM directory to a different directory enter the following command:

```
dsmc> restore sys:\tim\* sys:\newtim\
Restore function invoked.

Restoring          189 ITSC-NW1\SYS:/TIM/ITSC.SCR --> ITSC-NW1\SYS:/NEWTIM/ITSC.
SCR . Done
Restoring          475 ITSC-NW1\SYS:/TIM/ITSC2.SCR --> ITSC-NW1\SYS:/NEWTIM/ITSC
2.SCR . Done
Restoring          189 ITSC-NW1\SYS:/TIM/TEST.SCR --> ITSC-NW1\SYS:/NEWTIM/ITSC.
SCR . Done

Restore processing finished.
```

The command restores the contents of the directory to a directory called SYS:\NEWTIM. If this directory does not exist, it will be created, but with no trustee rights defined. The trustee rights and attributes for the old directory need to be created for the new directory. You can use this capability to restore entire directory structures or volumes to new locations.

4.1.3.2 Restoring Older Backup Versions

The restore operations discussed above restore the most recent backup version of the various files. ADSM with its management classes can maintain many backup copies of files. As files change and are backed up, ADSM keeps the number of backup versions specified in the management class. The last version is known as the *active* copy, and previous versions are known as *inactive* copies.

By default the NetWare ADSM client restores the most recent backup version or active copy of a file. You can select previous backup versions or inactive copies by using the INACTIVE and PICK options during a restore operation:

```
RESTORE  SYS:\TIM\ITSC.SCR      -PICK  -INACTIVE
```

The command displays a picking list of files that match the file name specified:

```
ADSM Scrollable PICK Window - Restore
     #    Backup Date/Time       File Size A/I  File
        ------------------------------------------------------------
     1. | 07/06/1994 14:18:53         189  A   ITSC-NW1\SYS:/TIM/ITSC.SCR
  X  2. | 07/06/1994 14:05:47         144  I   ITSC-NW1\SYS:/TIM/ITSC.SCR
        |
        |
        |
        0---------10--------20--------30--------40--------50--------60---
<U>=Up  <D>=Down  <T>=Top  <B>=Bottom  <R#>=Right  <L#>=Left
<G#>=Goto Line #  <#>=Toggle Entry  <+>=Select All  <->=Deselect All
<#:#+>=Select A Range <#:#->=Deselect A Range  <O>=Ok  <C>=Cancel
pick>
```

All backup versions available for that file are displayed. The active copy is marked with an A. Inactive copies are marked with an I. The file sizes and the date and time of backup are also displayed. You can then select and restore the required version.

4.1.3.3 Querying Backups

You can query which backups are available by using the QUERY BACKUP command:

```
dsmc> query backup sys:\tim\*
          Size      Backup Date       Mgmt Class A/I File
          ----      -----------       ---------- --- ----
           189  07/06/1994 14:18:53   DEFAULT     A  ITSC-NW1\SYS:/TIM/ITSC.SCR
           475  07/06/1994 13:46:11   DEFAULT     A  ITSC-NW1\SYS:/TIM/ITSC1.SCR
           189  07/06/1994 13:46:11   DEFAULT     A  ITSC-NW1\SYS:/TIM/TEST.SCR
```

This command displays the backups available for the specified path and file name. It also shows the management class that was used to back up the files. As with the restore command, you can use the INACTIVE option to list the active and inactive backup copies available.

4.1.4 Backup and Restore of the NetWare 3.1x Bindery

The bindery is a critical resource that must be backed up regularly. The NetWare ADSM client supports the backup and restore of the bindery using NetWare's SMS API (see 2.1.9, "Storage Management Services" on page 35). The bindery must be defined to ADSM in the DOMAIN option of the DSM.OPT file. ADSM treats the bindery as a separate type of file system.

4.1.4.1 Backup

You can back up the bindery by using either an incremental or selective backup. Whenever an incremental backup is performed, the bindery is always backed up. Because the bindery is a database that consists of three files, it does not have

attributes that can be used to determine whether it has been modified. ADSM always assumes that the bindery has been modified and backs it up as follows:

```
dsmc> incremental
Please enter password for node "NW1": ******

Session established with server ITSC: OS/2
  Server Version 1, Release 2, Level 0.0
  Server date/time: 07/07/1994 09:24:55   Last access: 07/06/1994 18:10:00.

Connecting to a NetWare 3.11 File System (ITSC-NW1).

Connected to ITSC-NW1.

Incremental backup of volume 'ITSC-NW1\SYS:'
ANS4102I ***** Processed       500 files *****
Successful incremental backup of 'ITSC-NW1\SYS:'

Incremental backup of volume 'ITSC-NW1\BINDERY:'
Normal File-->       2,048,000 ITSC-NW1\BINDERY:/BINDERY . Sent
Successful incremental backup of 'ITSC-NW1\BINDERY'
```

The bindery is backed up and sent to the ADSM server as a file called BINDERY. Before ADSM backs up the bindery, it closes it to ensure backup integrity. ADSM opens it after the backup is completed. The following messages appear on the NetWare server console:

```
 .
7/7/94 9:27am: 1.1.62 Bindery close requested by the SERVER
7/7/94 9:27am: 1.1.60 Bindery open requested by the SERVER
 .
```

To back up the bindery using selective backup, you issue the following command:

```
SELECTIVE BINDERY
```

Again, the bindery is first closed and then opened after the backup to ensure integrity. You receive the following messages after the SELECTIVE BINDERY command:

```
 .
Normal File-->       2,048,000 ITSC-NW1\BINDERY:/BINDERY . Sent
Selective Backup processing of 'bindery' finished with no failures.
 .
```

4.1.4.2 Restore

Restoring a backup copy of the bindery is simple; you enter the following command:

```
RESTORE BINDERY -PICK -INACTIVE
```

A list of bindery backup copies is displayed from which you can select the copy you want:

```
ADSM Scrollable PICK Window - Restore
     #    Backup Date/Time       File Size A/I  File
        ------------------------------------------------------------
     1. | 07/07/1994 09:37:42     2048000  A   ITSC-NW1\BINDERY:/BINDERY
     2. | 07/07/1994 09:27:47     2048000  I   ITSC-NW1\BINDERY:/BINDERY
        |
```

As with backup, the bindery is first closed and then opened after the restore. ADSM indicates that the bindery already exists and asks you to confirm that it should be restored.

Care should be taken when restoring the bindery. If an old version of the bindery is restored, there may be inconsistencies between this restored bindery and information in the NetWare file system, such as trustee assignments. A general recommendation is to run the NetWare BINDFIX utility after restoring a bindery. This utility will report on and fix any such inconsistencies.

4.1.5 Backup and Restore of the NetWare 4.x Directory

The ADSM release 2 NetWare client supports backup and restore of the directory on NetWare version 4 servers. The NetWare server must have TSANDS.NLM loaded (see 3.1.1, “Updating the Server AUTOEXEC.NCF File” on page 44).

NetWare version 4 differs from version 3 in that there are separate TSAs for the directory and file systems. On NetWare version 3.1x a single TSA backs up both the bindery and file systems.

When installing the ADSM client on a NetWare version 4 server, the DOMAIN option must include an entry for the directory, if it is to be backed up (see 3.1.2, “Updating the ADSM Options File” on page 45).

4.1.5.1 Backup

You can back up the NetWare directory using an incremental or selective backup. You can back up either a whole directory or a portion of the directory. The directory is a tree structure consisting of a series of organizations, organization units, and common names. ADSM can back up these individual components of the directory.

To perform an incremental backup of the complete directory, enter the following command:

```
INCREMENTAL DIRECTORY/FULL
```

Figure 17 on page 38 presents an example of a directory. In that example, you could back up the container organization unit (OU) called STORAGE separately. Use the following command to perform an incremental backup of just the STORAGE portion of the directory:

```
INCREMENTAL DIRECTORY/.OU=STORAGE
```

You can also use selective backups on the full directory.

Careful thought needs to be given to how you back up the directory. You need to consider the advanced features of the NetWare directory, such as the ability to replicate. Directory replication, if used, provides a form of online mirroring between NetWare version 4 servers.

In the directory replication scenario, using ADSM as well needs careful planning. There is little benefit to be gained from using ADSM to back up a NetWare version 4 directory, if it is only a replica of a directory on another server. Obviously the directory should be backed up somewhere, but not necessarily on every NetWare server.

4.1.5.2 Restore

The process used to restore the NetWare directory is the same as that used for other files. You can perform a full restore or partial restore of a container. To restore the STORAGE container in our example, enter the following command:

```
RESTORE DIRECTORY/.OU=STORAGE
```

If directory replication is being used, the restored portion of the directory will be replicated on the other NetWare servers.

4.2 Archive and Retrieving Data

The objective of ADSM′s archive and retrieval functions is long-term vital data retention, rather than backup. Files archived are retained for the length of time specified in the management class′s archive copy group and then automatically deleted. Files archived are not deleted from the file system. The two archive commands, ARCHIVE and RETRIEVE, are used to archive and retrieve files.

ADSM archive is not an automatic space management function. If you use archive to reclaim space on a file system, you must manually delete the files after they have been archived. There is no file version control with archive, unlike with backup. The management classes for archive only control the retention period of the archive copies. There is no limit to how many copies of an individual file can be archived.

4.2.1 Archiving

You can archive files using the ARCHIVE command. As the objective is to provide a facility for long-term storage of data, some method of identifying the files in addition to their file name is needed. ARCHIVE allows a text description to be attached to a file when archived. In the following example, the SYS:\TIM\ITSC.SCR file is archived, and a description is attached to it with the DESCRIPTION option:

```
dsmc> archive sys:\tim\itsc.scr -description="example archive for redbook"
Archive function invoked.

Archiving-->               189 ITSC-NW1\SYS:/TIM/ITSC.SCR . Sent
Archive processing of 'sys:\tim\itsc.scr' finished with no failures.
```

Pattern matching can also be used for archives. A file specification of ITSC*.* will archive all files in that directory that match the pattern. In this case two files matched the pattern and were archived:

```
dsmc> archive sys:\tim\itsc*.* -description="archive with * wildcard"
Archive function invoked.

Archiving-->               189 ITSC-NW1\SYS:/TIM/ITSC.SCR . Sent
Archiving-->               475 ITSC-NW1\SYS:/TIM/ITSC2.SCR . Sent
Archive processing of 'sys:\tim\itsc*.scr' finished with no failures.
```

4.2.2 Retrieving

You can retrieve archived data with the RETRIEVE command. With retrieval, you can search against both the file name and the description. A list of selected files is provided if you use the following command:

```
RETRIEVE  SYS:\TIM\*     -DESCRIPTION="*WILD*"  -PICK
```

A file can then be selected and retrieved. If the file still exists, you are prompted to replace it. Files that have been archived from one directory can be retrieved to another with a different file name. Here is the list that is displayed for a retrieve request:

```
ADSM Scrollable PICK Window - Retrieve

     #    Archive Date/Time      File Size   File
        ------------------------------------------------------------
     1. | 07/07/1994 14:40:41         189   ITSC-NW1\SYS:/TIM/ITSC.SCR
     2. | 07/07/1994 14:40:41         475   ITSC-NW1\SYS:/TIM/ITSC2.SCR
        |
        |
```

4.2.3 Deleting Archive Data

There is one function that is unique to archive: the ability to delete individual archived copies of files. No such function exists with the backup function. The backup file version control implemented in the ADSM management classes limits the number of backup copies that are held. No such controls exist for archive data.

There is no limit to the number of archived copies that can be held. Combined with the long retention periods typical of archived data, a mechanism is provided to perform housekeeping. You can select a list of archived copies and choose the copies to be deleted by using the DELETE ARCHIVE command:

```
DELETE  ARCHIVE  SYS:\TIM\ITSC*.*     -PICK
```

This command displays a picking list of matching archive files from which you can choose the archive copies to be deleted:

```
ADSM Scrollable PICK Window - Archive Delete

     #    Archive Date/Time      File Size   File
        ------------------------------------------------------------
     1. | 07/07/1994 15:15:23         189   ITSC-NW1\SYS:/TIM/ITSC.SCR
     2. | 07/07/1994 15:15:21         189   ITSC-NW1\SYS:/TIM/ITSC.SCR
     3. | 07/07/1994 14:40:41         189   ITSC-NW1\SYS:/TIM/ITSC.SCR
     4. | 07/07/1994 15:15:23         475   ITSC-NW1\SYS:/TIM/ITSC2.SCR
     5. | 07/07/1994 15:15:21         475   ITSC-NW1\SYS:/TIM/ITSC2.SCR
     6. | 07/07/1994 14:40:41         475   ITSC-NW1\SYS:/TIM/ITSC2.SCR
        |
```

4.3 Running Scheduled Operations

An alternative to the backup and archive methods discussed above is to run scheduled backups or archives. The schedules are defined on the ADSM server and invoked at the ADSM client. Restore or retrieval of files cannot be scheduled.

For the NetWare ADSM client to run a schedule it must be started in a scheduled mode. This is done by loading the DSMC NLM with a parameter called SCHEDULE. To start the NetWare ADSM scheduled client, enter the following at the NetWare server console:

```
LOAD DSMC SCHEDULE
```

The ADSM client will load, connect to the ADSM server, and prompt for the ADSM client password. The ADSM client will then request details of its next scheduled operation. Here is an example of the ADSM screen after the ADSM client has received its schedule:

```
Session established with server ITSC: OS/2
  Server Version 1, Release 2, Level 0.0
  Server date/time: 07/07/1994 15:45:40   Last access: 07/07/1994 15:44:00

Querying server for next scheduled event.
Next operation scheduled:
------------------------------------------------------------
Schedule Name:         NW_DAILY
Action:                Incremental
Objects:
Options:
Server Window Start:   16:00:00  on 07/07/1994
------------------------------------------------------------
Command will be executed in 15 minutes.
```

Note: This is an example of the client polling method of central scheduling. The two methods, client polling and server prompted, are discussed in 1.3.7.2, “Scheduling Methods” on page 19.

The ADSM server has returned the next scheduled operation for this client. The ADSM client now waits until it is time to start the incremental backup. When it is time, the ADSM client reconnects to the ADSM server and starts the scheduled backup:

```
Executing scheduled command now.
Session established with server ITSC: OS/2
  Server Version 1, Release 2, Level 0.0
  Server date/time: 07/07/1994 16:00:34   Last access: 07/07/1994 15:45:00
  .
  .
```

On completion of the operation the ADSM client sends the results to the ADSM server and requests details of the next scheduled operation. The ADSM client now waits to perform its next schedule. It periodically queries the ADSM server to confirm that the schedule is still valid:

```
Scheduled event 'NW_DAILY' completed successfully.
Sending results for scheduled event 'NW_DAILY'.
Results sent to server for scheduled event 'NW_DAILY'.

Session established with server ITSC: OS/2
  Server Version 1, Release 2, Level 0.0
  Server date/time: 07/07/1994 16:01:43   Last access: 07/07/1994 16:00:

Querying server for next scheduled event.
Next operation scheduled:
------------------------------------------------------------
Schedule Name:         NW_DAILY
Action:                Incremental
Objects:
Options:
Server Window Start:   16:00:00  on 07/08/1994
------------------------------------------------------------
Schedule will be refreshed in 12 hours.
```

If server prompted scheduling is used, when the scheduled ADSM client is started the following information appears on the ADSM screen:

```
Querying server for next scheduled event.
Next operation scheduled:
------------------------------------------------------------
Schedule Name:         NW_DAILY
Action:                Incremental
Objects:
Options:
Server Window Start:   16:00:00  on 07/08/1994
------------------------------------------------------------
Waiting to be contacted by the server.
```

With server prompted scheduling, rather than the NetWare server starting the operation, as is done with client polling scheduling, the ADSM client waits to be contacted by the ADSM server and told to start an operation.

4.3.1 Scheduling Related Client Options

A number of ADSM options in the DSM.OPT file relate to scheduled operations and how they are performed. All of these options have default values, which are fine when the NetWare client is first installed. However, you need to review them when planning ADSM schedules. The options are:

MAXCMDRETRIES This option specifies the number of retries the NetWare ADSM client will make if a scheduled operation fails. For example, if the client cannot contact the ADSM server, it will retry the number of times specified by this option. The default value is 2.

QUERYSCHEDPERIOD This option is used to define the number of hours that elapse between the times that the ADSM client queries the ADSM server to check the scheduled operations.

This option is useful if the ADSM administrator changes the time when the schedule starts or the operation that ADSM is going to perform. The option allows the ADSM client to check with the server frequently so as to have the most current information after a changed schedule.

This option is used only if client polling scheduling is used. The default value is 12.

RETRYPERIOD This option is associated with MAXCMDRETRIES. It defines how many minutes elapse between retries. The default is 12 minutes.

SCHEDLOGNAME The NetWare ADSM client has an error log to record any errors that occur during scheduled operations. This option can be used to specify the name of the log and the directory where it is created. By default the error log file is called DSMSCHED.LOG and is put in the directory where the ADSM client code is installed.

SCHEDMODE This option determines whether client polling or server prompted scheduling is used. The default is client polling.

Note: Server prompted scheduling is supported only for ADSM clients that use TCP/IP connectivity.

TCPCLIENTADDRESS This option is associated with server prompted scheduling. It defines the TCP/IP address that the ADSM server will use to contact the ADSM client. There is no default for this option.

TCPCLIENTPORT This option is also associated with server prompted scheduling. It defines the port that the ADSM server will use when it establishes contact with TCP/IP on the NetWare server. You can use this option to have a different port number for scheduled operations from the one normally used by the NetWare ADSM client for nonscheduled operations. The default is 1501.

Chapter 5. Exploiting the ADSM NetWare Client

In this chapter we look at how the ADSM client on a NetWare server can be used to full advantage. Some of the topics discussed here are ADSM related, others are functions of NetWare that can be used to full advantage in conjunction with the ADSM client. We cover running multiple ADSM clients on a single server and managing multiple NetWare servers and present some thoughts on NetWare server recovery and disaster recovery.

5.1 Multiple ADSM Clients on a Single NetWare Server

In our discussion on setting up the NetWare ADSM client, it should have become apparent that ADSM is configured as a simple client/server model. A single ADSM client communicates with a single ADSM server, using the defined communications method.

There are situations where it is desirable to use multiple ADSM clients on a single NetWare server, for example, if the ADSM client will be running scheduled backups at the same time that other NetWare ADSM storage administration functions are being done.

The ADSM client uses the DSM.OPT file to configure how it will operate and communicate with the ADSM server. Within the DSM.OPT file for the NetWare client only one communications protocol and path to an ADSM server can be defined.

When the ADSM client is started with the LOAD DSMC command, the DSM.OPT file that is in the same directory as the DSMC.NLM is used. With NetWare there is no capability to reference another OPT file as can be done with some other ADSM client platforms. However, DSMC is a reentrant module. It can be loaded multiple times on the same NetWare server. At the NetWare server console LOAD DSMC can be entered as often as you like. Every time it is entered, another ADSM client session is started.

It is possible for you to have an infinite number of client sessions running on the same NetWare server, all performing different operations. You can press the <CTRL> and <ESC> keys at the NetWare server console to display a list of the available console sessions that you can select. Below is an example of the console session when LOAD DSMC has been entered three times:

```
Current Screens

     1. System Console
     2. Monitor Screen
     3. ADSTAR Distributed Storage Manager
     4. ADSTAR Distributed Storage Manager
     5. ADSTAR Distributed Storage Manager

Select screen to view:
```

Three ADSM client sessions are running, all connected to the same ADSM server. At the ADSM server three sessions would be seen from the same NetWare ADSM client. ADSM allows multiple simultaneous sessions from the same client.

The ability to run mulitple ADSM clients enables scheduled client sessions that have been started using the LOAD DSMC SCHEDULE command to be left running while NetWare administrators or other ADSM users start or stop other ADSM sessions.

Unfortunately there is no easy way of identifying which of the three sessions above is the scheduled session. You have to select each session in turn to see what it is doing. When you need to stop one or more of the three sessions, you have two alternatives:

- Enter UNLOAD DSMC at the NetWare server console.
- Enter QUIT from the ADSM screen.

If multiple client sessions are running, the first option is to be avoided. In the example above, entering UNLOAD DSMC at the NetWare server console will unload all three ADSM sessions. The NetWare UNLOAD command cannot differentiate between the three DSMC NLMs that are loaded. It just sees three copies of the NLM and unloads all three. Thus, if multiple client sessions are running, always stop them by entering QUIT from the ADSM session at the DSMC> prompt.

5.1.1 Multiple ADSM Clients with Different ADSM Servers

There also will be situations where it might be desirable to use the NetWare ADSM client with more than one ADSM server. For example, if the ADSM server is located in the same building as the ADSM client, it might be desirable to have another offsite ADSM server for disaster recovery purposes.

If you want to run multiple ADSM sessions to different ADSM servers, you need multiple DSM.OPT files. When you LOAD DSMC, it will always use the DSM.OPT file in the same directory where DSMC.NLM is stored. There is no way of changing this with the NetWare ADSM client.

If you require separate DSM.OPT files to connect to different ADSM servers, you also need separate DSMC.NLMs and the other associated files. These can be loaded separately using a different path statement (see Figure 23 on page 93).

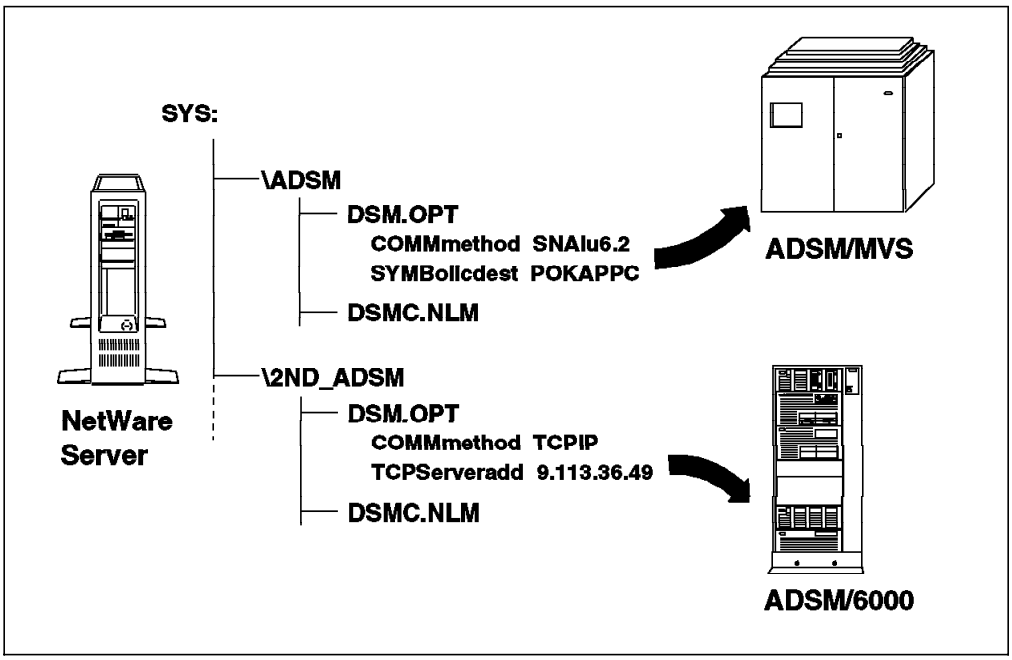


Figure 23. NetWare ADSM Client with Multiple ADSM Directories

A separate directory, 2ND_ADSM, has the complete set of ADSM files copied from the ADSM directory. In 3.1.1, "Updating the Server AUTOEXEC.NCF File" on page 44 we add a search path statement to the NetWare AUTOEXEC.NCF for the SYS:\ADSM directory. Therefore whenever LOAD DSMC is entered at the server console, the DSMC.NLM that is loaded is the one located in the SYS:\ADSM directory. The DSMC.NLM uses the DSM.OPT located in this same directory and connects to the MVS ADSM server using APPC connectivity.

In this example a second ADSM server, ADSM/6000, can be accessed by modifying the SYS:\2ND_ADSM\DSM.OPT. The COMMMETHOD can be set to TCPIP and the appropriate server address entered. The copy of the DSMC.NLM in this directory, along with this new DSM.OPT file, can be executed by entering the following command:

```
LOAD SYS:\2ND_ADSM\DSMC.NLM
```

Using the LOAD command with a fully qualified path name ignores the NetWare SEARCH path statements and loads from the directory specified. This will start a session with the ADSM/6000 server. The above command could also be put into a NetWare NCF file so that it could be easily executed.

Using this technique it is possible to run multiple concurrent sessions to different ADSM servers. There is an overhead in that multiple copies of the NetWare ADSM client code must be maintained. Also care must be taken when working with ADSM in this type of environment; you must be sure that you are using the ADSM session you think you are using.

5.2 Managing Multiple NetWare Servers with an ADSM Client

With NetWare SMS, a storage management application can access remote NetWare servers using the SMDR and TSA components. A single NetWare ADSM client can manage multiple NetWare servers in this way.

5.2.1 Configuration and Use

In 3.1, “Installing the ADSM Client” on page 43 we cover installing the SMS modules. These are the SMDR and TSA NLMs required for the ADSM client to access a NetWare server. The modules are initially installed on the same NetWare server as the ADSM client to enable the ADSM client to back up and archive data from the local file system. (Local is the server on which the NetWare ADSM client runs.)

The SMDR and TSA NLMs can also be loaded on a different NetWare server and accessed by the ADSM client from the original NetWare server. The requirements for the NetWare ADSM client to manage a remote server are:

- The NetWare servers must be able to communicate with each other. If they are on separate segments of the LAN, the gateways must support routing of IPX/SPX traffic between the segments.
- The SMDR and TSA NLMs must be loaded on all NetWare servers that the ADSM client manages.
- The DSM.OPT file must be configured to reflect all NetWare servers being managed.

Figure 24 on page 95 illustrates how two NetWare servers can be configured to enable one to manage the other. In the example two NetWare servers are connected to a token-ring LAN. Their server names are ITSC-NW1 and ITSC-NW2. They are NetWare version 3.11 servers, and each has a SYS: volume. The ITSC-NW1 server has the ADSM client installed and configured.

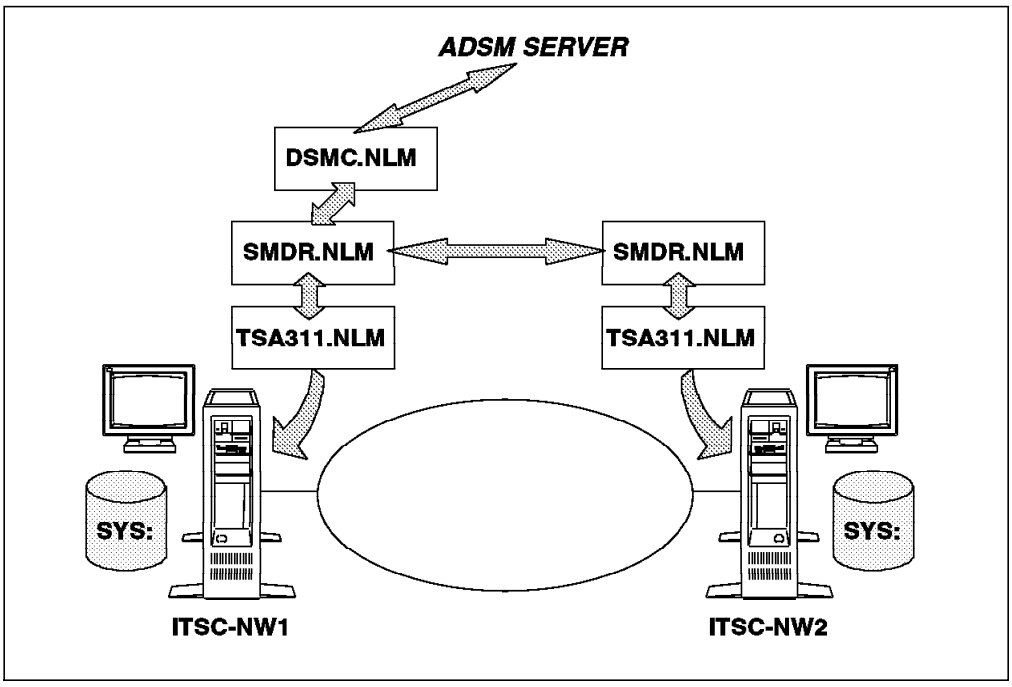


Figure 24. Two NetWare Servers with a Single ADSM Client

The ITSC-NW2 server must have the following NetWare SMS modules installed and loaded:

- SMDR31X and TSA311 NLMs for a NetWare version 3.11 server
- SMDR31X and TSA312 NLMs for a NetWare version 3.12 server
- SMDR, TSA400, and TSANDS for a NetWare version 4.01 server. The TSANDS is only required if the NetWare Directory Services will be backed up.

The NetWare SMS modules can be installed from the ADSM client diskette, or they can be copied across the LAN from another NetWare server if they have already been installed there. It is also recommended that their AUTOEXEC.NCF file on the remote NetWare server be updated with a LOAD statement to load the required TSAs.

```
LOAD TSA311
```

Loading the TSA module automatically loads the SMDR if it was not previously loaded. In this example both servers are NetWare version 3.11.

5.2.1.1 Updating DSM.OPT

The ADSM client on the ITSC-NW1 server must be configured to connect to the remote NetWare server. Add the following DOMAIN statements in the DSM.OPT file:

```
DOMAIN      ITSC-NW1\SYS:  ITSC-NW1\BINDERY
DOMAIN      ITSC-NW2\SYS:  ITSC-NW2\BINDERY
```

The first domain statement applies to the local NetWare server, and the second, to the remote server, ITSC-NW2. With these entries the ADSM client on

ITSC-NW1 will back up the SYS: volume and the bindery on both NetWare servers.

Additionally the INCLUDE/EXCLUDE lists must be updated. These lists can contain fully qualified entries with the NetWare server name included. If a server name is not specified in the INCLUDE/EXCLUDE statements, the entry defaults to the local NetWare server's file systems specified in the DOMAIN statement:

```
EXCLUDE        SYS:\...\*
```

This statement excludes all files in all directories on the SYS: volume on the ITSC-NW1 server. However, all files and directories on the remote ITSC-NW2 server will be included. Therefore if you want a general exclude statement for all SYS: volumes, you would want to put one of the following in the DSM.OPT file:

```
EXCLUDE       ITSC-NW1\SYS:\...\*
EXCLUDE       ITSC-NW2\SYS:\...\*
or
EXCLUDE             *\SYS:\...\*
```

The fully qualified *Server/Volume:/path* must be specified. Pattern matching characters such as an asterisk (*) can be used as wildcards for any of the elements in either INCLUDE or EXCLUDE statements. This rule also applies for the bindery or directory, if you are backing them up.

The ADSM node name specified in the DSM.OPT file is not affected by including remote NetWare servers in the configuration. The ADSM client still has a single node name, which is the local NetWare server where the ADSM client is installed. The ADSM server sees only one ADSM client but with additional file systems being managed. The ADSM server has no knowledge that remote NetWare servers are being managed.

5.2.1.2 Invoking the ADSM client

You can use the NetWare ADSM client to manage a remote NetWare server in exactly the same way you used it to manage a single local server. At the NetWare console or a remote console session, start the ADSM client by entering LOAD DSMC. You can then enter ADSM commands that process on the local and remote NetWare servers. For example, an incremental backup will back up all volumes and binderies on both NetWare servers:

```
dsmc> incremental

Connecting to a NetWare 3.11 File System (ITSC-NW1).

Connected to ITSC-NW1.

Incremental backup of volume 'ITSC-NW1\SYS:'
ANS4102I ***** Processed       500 files *****
Successful incremental backup of 'ITSC-NW1\SYS:'

Incremental backup of volume 'ITSC-NW1\BINDERY:'
Normal File-->       2,048,000 ITSC-NW1\BINDERY:/BINDERY . Sent
Successful incremental backup of 'ITSC-NW1\BINDERY'

Connecting to a NetWare 3.11 File System (ITSC-NW2).

Please enter NetWare user for "ITSC-NW2": tim

Please enter the password on "ITSC-NW2" for NetWare user "TIM": *****

Connected to ITSC-NW2.

Incremental backup of volume 'ITSC-NW2\SYS:'
Directory-->                 0 ITSC-NW2\SYS:/ . Sent
  .
  .

Successful incremental backup of 'ITSC-NW2\SYS:'

Incremental backup of volume 'ITSC-NW2\BINDERY:'
Normal File-->       2,048,000 ITSC-NW2\BINDERY:/BINDERY . Sent
Successful incremental backup of 'ITSC-NW2\BINDERY'
```

This is an extract of what you see at the console of the ITSC-NW1 NetWare server. The ADSM client starts by backing up the local SYS: volume and bindery on ITSC-NW1. When the ITSC-NW1 backup completes, ADSM connects to the SMDR and TSA modules on the remote ITSC-NW2 server. The incremental backup user is prompted for a NetWare userid and password on the ITSC-NW2 server. This userid must have sufficient authority to access all files to be backed up.

If you use the NWPWFILE option, the userid and password will be stored and encrypted in another .PWD file in the ADSM directory of the local ITSC-NW1 server. This ensures that future operations that require access to ITSC-NW2 will not prompt for the userid. Once connected, the incremental backup proceeds to back up the SYS: volume and bindery on the ITSC-NW2 server.

With this type of configuration you can perform all of the normal ADSM tasks, but now with further choice of source and destination for the data residing on multiple NetWare servers. For example, a file backed up on one server can be restored to another:

```
dsmc> restore itsc-nw2\sys:\system\autoexec.ncf itsc-nw1\sys:\system\nw2\auto
      exec.nw2
Restore function invoked.

Connecting to a NetWare 3.11 File System (ITSC-NW1).

Connected to ITSC-NW1.

Restoring          194 ITSC-NW2\SYS:/SYSTEM/AUTOEXEC.NCF --> ITSC-NW1\SYSTE
M/NW2/AUTOEXEC.NW2 . Done

Restore processing finished.
```

This example is of a backup copy of the AUTOEXEC.NCF file from the ITSC-NW2 server being restored to a new directory on the ITSC-NW1 server, where it will be kept in case of a system failure on the ITSC-NW2 server.

5.2.2 Operational Considerations

The ability to access a remote NetWare server is very useful and provides great configuration flexibility. It enables multiple NetWare servers to be managed from a single NetWare server with the ADSM client and host connectivity installed and configured.

Remote NetWare server access may be particularly useful where APPC connectivity is to be used. It is quite common for a run-time version of NetWare with NetWare for SAA to be installed as a communications gateway to ensure that the NetWare for SAA processor overhead does not impact the performance of NetWare file servers.

The architecture of NetWare demands that NetWare for SAA and ADSM run on the same physical NetWare server. In this situation the ability to run ADSM on the NetWare for SAA server and manage other NetWare file servers provides great flexibility.

The disadvantage of managing multiple NetWare servers is the amount of data that can be owned by a single ADSM client. The ADSM server will logically see a single ADSM client with a large number of file spaces. If, in the example, a QUERY FILESPACE command is entered, the following will be seen:

```
dsmc> query filespace
Session established with server ITSC: OS/2
  Server Version 1, Release 2, Level 0.0
  Server date/time: 07/10/1994 11:19:45   Last access: 07/10/1994 10:58:00

Num     Last Incr Date      Type       File Space Name
---     --------------      ----       ---------------
  1   07/10/1994 10:59:09   NWBINDRY   ITSC-NW1\BINDERY:
  2   07/10/1994 10:59:05   NTWFS      ITSC-NW1\SYS:
  3   07/10/1994 10:59:57   NWBINDRY   ITSC-NW2\BINDERY:
  4   07/10/1994 10:59:55   NTWFS      ITSC-NW2\SYS:
```

FILESPACE is the ADSM term for the file system's repository on the ADSM server where backup and archive data is held. File spaces are owned by ADSM clients. For the NetWare ADSM client, a file space corresponds to a NetWare volume and bindery or directory on a NetWare server.

This ADSM server has four file spaces owned by the ADSM client. This single ADSM client has the responsibility and task of backing up and restoring this data for all of the NetWare servers it manages. This could become an operational bottleneck.

It is possible for one NetWare ADSM client to manage every NetWare server in an enterprise. However, this single NetWare server with the ADSM client will become a single point of failure and an operational and performance bottleneck. If a single ADSM client managing multiple NetWare servers is to be used, a balance needs to be achieved with a sensible ratio of ADSM clients to NetWare servers.

5.2.3 Performance Considerations

The performance implication of managing multiple NetWare servers with a single ADSM client also should be understood. When the NetWare ADSM client performs an incremental backup, it is a serial process. It starts backing up the volumes on the local server first, including the bindery or directory. When it has completed those backups, it then connects to the remote NetWare servers and backs them up.

This process is performed in the sequence in which the DOMAIN statements are defined in the DSM.OPT file. The greater number of remote systems that are managed, the longer it will take. In a situation where there is a critical time window for the backups, this might be an issue.

Network performance also must be considered. In the configuration with a single ADSM client managing multiple NetWare servers, data is transferred from the NetWare server with the ADSM client, to the ADSM server, using the chosen communications protocol: APPC, IPX/SPX, PWSCS, or TCP/IP. The communications protocol can be tuned to give optimum throughput. This tuning benefits the data that is being backed up on the local server. However, the communications protocol used to connect to the remote servers is IPX/SPX.

The remote NetWare servers could be across bridges or gateways that could seriously degrade overall ADSM performance. A situation could arise where the NetWare server with the ADSM client has a fast connection to the ADSM server. However, the speed at which backups can be performed is determined by the speed at which ADSM can obtain data from remote servers.

The final factor that must be understood in this environment is compression. ADSM optionally compresses data at the client system before it is transmitted across the network to the ADSM server. The use of compression is set in the ADSM client by adding the following to the DSM.OPT file:

```
COMPRESSION    YES
```

Compression has two potential benefits:

- Decreasing the amount of data that is transmitted across the network, with a resulting reduction in elapsed backup time.
- Reducing the amount of physical storage pool capacity required on the ADSM server.

In an environment where remote NetWare servers are managed, compression may only produce limited benefit. Compression is performed at the NetWare server where the ADSM client is running, not at the remote NetWare server. Data is sent decompressed from the remote server to the local server, where it is compressed and sent to the ADSM server. In this environment, compression will probably have little impact on overall backup or archive elapsed time.

5.2.4 Summary

The ability to manage remote NetWare servers with a single NetWare ADSM client is attractive. It provides a great degree of configuration flexibility. However, the operational and performance considerations need to be understood and weighed against the configuration benefits.

5.3 NetWare Server Recovery with ADSM

Our discussion in Chapter 4, “Using the ADSM NetWare Client” on page 77 focused on backing up and restoring a file, directory, or NetWare volume. This is the typical day-to-day activity that is performed with the NetWare ADSM client. The assumption was that the NetWare server was functional and that ADSM was operational. Situations will arise, however, where the NetWare server fails and the ADSM client, if running on that server, will be unusable.

Planning an ADSM configuration carefully can minimize the possibility of serious data loss due to the failure of a NetWare server. Certain failures cannot be protected against with ADSM. A disk hardware problem that demounts the server’s SYS: volume is catastrophic. If that NetWare server is also running the ADSM client, ADSM cannot be used.

On a NetWare server the SYS: volume is a single point of failure. Without SYS:, a NetWare server is incapable of performing any tasks. In the worst case, if the problem cannot be resolved and the volume repaired, it might mean that NetWare has to be reinstalled. If that server was being used for the ADSM client, ADSM and all connectivity software also would have to be reinstalled and configured.

The DOS partition from which NetWare is started is inaccessible once NetWare has started. The NetWare ADSM client cannot access that DOS partition. Thus the DOS system is a critical single point of failure on a NetWare server.

The NetWare ADSM client is very flexible and can be configured in many ways. In planning an ADSM implementation for NetWare, this configuration flexibility should be exploited. In an environment where there are multiple NetWare servers, it may or may not be necessary to run the ADSM client on every NetWare server. In the sections that follow we look at a possible NetWare server recovery scenario.

5.3.1 Using Multiple NetWare ADSM Clients

Figure 25 shows a fairly typical configuration with two NetWare servers, each with a NetWare ADSM client. Each NetWare server performs its own regular backups. Both are connected to the same ADSM server.

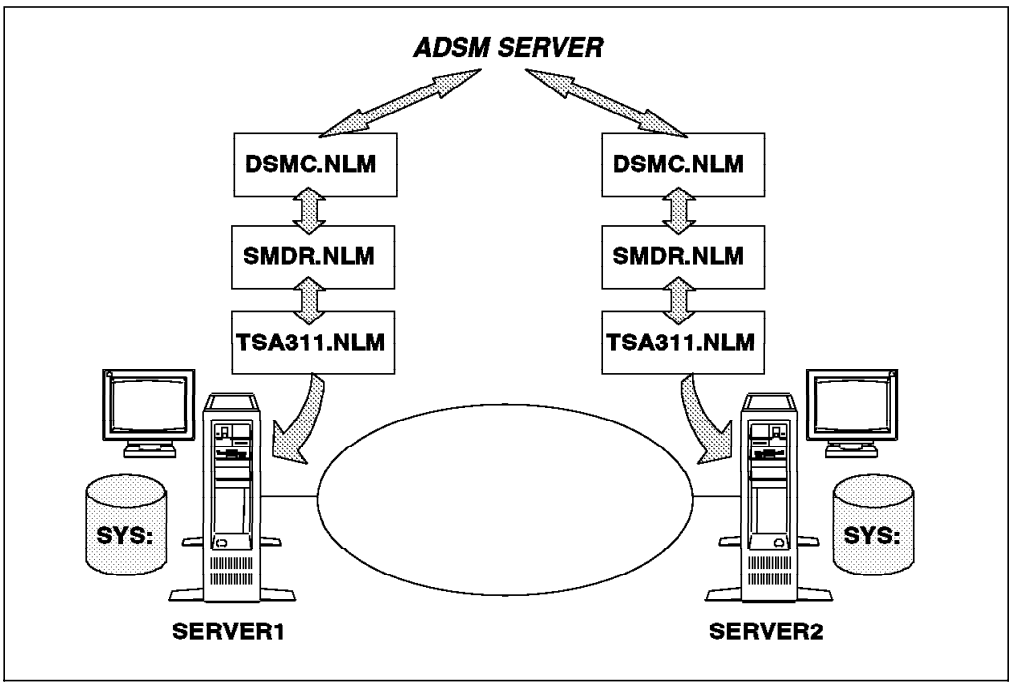


Figure 25. NetWare Overview

Restoring files or directories is a straightforward process from the NetWare server. If a serious problem occurs with a server, however, it is a more difficult situation to manage. Let us take a worst case example: SERVER2 develops a hardware fault, catches fire, and is destroyed. There is little alternative but to install a new PWS and rebuild the NetWare system.

With ADSM, the NetWare system could be recovered in two ways:

- Reinstall NetWare, the ADSM client, and any additional connectivity software required to get the ADSM client working again. Reconfigure ADSM and the host connectivity. Once an ADSM session has been reestablished with the ADSM server, start to restore the other data on the NetWare server. In an environment where there is only one NetWare server and ADSM client this is all that can be done.
- Reinstall a NetWare system and use the cross client restore capability of the NetWare ADSM client to rebuild the server, as described below.

ADSM provides the capability for an ADSM client to authorize another client to access its backup or archive data on the ADSM server. This authority can be global for all data, or a limited subset of the backup and archive data can be authorized.

Cross client restore only works between like ADSM clients. Data backed up by the NetWare ADSM client can be restored only by another NetWare ADSM client. It cannot, for example, be restored by the OS/2 or DOS ADSM client.

Authorizations are defined by the ADSM client that owns the data and authorizes other clients to access it. In the example in Figure 25, the ADSM client on the SERVER2 NetWare server could authorize SERVER1 to access all backup data owned by SERVER2 by entering the following ADSM command:

```
SET ACCESS BACKUP * SERVER1
```

The QUERY ACCESS command can then be entered to confirm the settings. The following would be observed at the ADSM screen:

```
dsmc> set access backup * server1
ANS4198I 'Set Access' command successfully completed
dsmc> query access
Type     Node        User        Path
----     ----------------------------
Backup   SERVER1     *           *

ANS4198I 'Query Access' command successfully completed
```

The ADSM client on SERVER1 can now be used, if necessary, to restore data on SERVER2. To restore data from another ADSM client, the -FROMNODE= option is used with the RESTORE command. This identifies the owning ADSM client node. The following example is of the ADSM client on SERVER1 restoring a file previously backed up by SERVER2:

```
dsmc> restore -fromnode=server2 server2\sys:\fl\oj.drw
Restore function invoked.

Connecting to a NetWare 3.11 File System (SERVER2).

Please enter NetWare user for "SERVER2": supervisor

Please enter the password on "SERVER2" for NetWare user "SUPERVISOR": *****

Connected to SERVER2.

File SERVER2\SYS:/FL/OJ.DRW already exists, do you want to replace it (Yes/
No) yes
Restoring          207 SERVER2\SYS:/FL/OJ.DRW . Done

Restore processing finished.
```

In this example the ADSM client on SERVER1 has restored a file previously backed up by the ADSM client on SERVER2. The ADSM user is prompted for a userid and password to access SERVER2 when writing the restored file. In this example the file has been written back to its original location on SERVER2.

The NetWare ADSM client on SERVER2 does not need to be running for this restore to occur. The only requirement is that the SMS modules, the SMDR and TSA NLMs appropriate for the server version, be loaded. Also the LAN must support IPX/SPX protocols between servers.

In our hypothetical scenario, where SERVER2 has burst into flames, its data can now be recovered. A new server machine has to be installed and NetWare

reinstalled. The new server should be configured with the same NetWare server name, SERVER2, as before.

Once the new SERVER2 is operational, the only additional task required is to load the appropriate SMS modules, SMDR and TSA. At this point the NetWare ADSM client on SERVER1 can access this server and start restoring the backup data from the original SERVER2. There are two restore approaches:

- The entire backup for SERVER2 can be restored by SERVER1. As this process uses IPX/SPX protocols between the servers to transfer the data, this might not be acceptable. The speed of restore will probably be slower than using the ADSM client natively on the SERVER1 NetWare server. Also the LAN traffic might adversely impact other users.
- Just enough data can be restored on SERVER2 to get the original ADSM client working again. The ADSM client can then be started on SERVER2 and the rest of the data restored. This approach is potentially faster than the first approach and might not impact other LAN users as much. However, the impact will depend on the way in which the NetWare ADSM client communicates with the ADSM server.

In summary, one can recover from the loss of a NetWare server. If it is a serious recovery problem, NetWare might have to be reinstalled. Nothing can be done with ADSM to avoid that; however, the ADSM configuration could be such that all NetWare server data can be recovered subsequently. The key recommendations for this scenario are:

- Run multiple ADSM clients on different NetWare servers whenever possible.
- On all NetWare ADSM clients, authorize every other NetWare ADSM client access to the server backup data.
- Always run the SMS modules, SMDR and TSA, on all NetWare servers. This will enable any NetWare ADSM client to access that server if required.

5.3.2 Multiple ADSM Clients with Remote NetWare Servers

Cross client restore between ADSM clients and the ability to manage remote NetWare servers with the NetWare ADSM client provide a very high degree of flexibility and resilience. Figure 26 on page 104 illustrates a system configuration that exploits both capabilities.

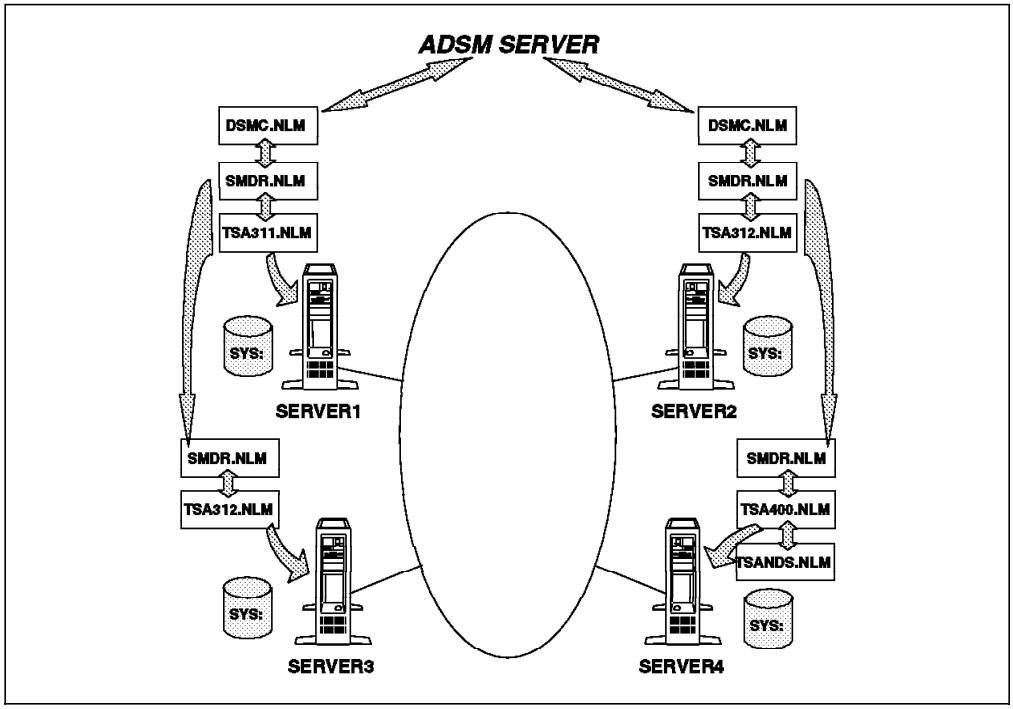


Figure 26. Multiple ADSM Clients Managing Remote NetWare Servers

In this configuration there are four NetWare servers, SERVER1, SERVER2, SERVER3, and SERVER4. The servers are a variety of NetWare versions 3.11, 3.12, and 4.01. They are connected to a LAN and can communicate with each other. The LAN could be multisegment, but as long as the gateways route IPX/SPX protocols, the configuration is valid.

Two of the NetWare servers are also ADSM clients. These are SERVER1 and SERVER2. These two ADSM clients manage the data on their own file systems and on a remote system. SERVER1 manages SERVER3, and SERVER2 manages SERVER4.

Each ADSM client authorizes the other to access its backup data by entering the following ADSM command:

```
SET ACCESS BACKUP * *
```

This command allows either ADSM client to restore any backup data from the other ADSM client.

The ADSM client on SERVER2 can now restore data on SERVER3 with the following command:

```
RESTORE -FROMNODE=SERVER1 SERVER3\SYS:\FL\*.*
```

This command restores the contents of the SYS:\FL directory on SERVER3, which has been backed up by the ADSM client on SERVER1.

This configuration has a number of possible benefits:

- Data on any NetWare server can be restored by any ADSM client.
- If a NetWare server with the ADSM client goes down, its data can be recovered from another NetWare server with the ADSM client.
- There are no single points of failure in the ADSM client configuration.

- If required, each ADSM client could be configured to back up to multiple ADSM servers.
- Ad hoc restores can be performed from a single point. A NetWare administrator can recover data to any NetWare server from one server console or remote console session without the need to log in to the target server.
- The ADSM client could be run on a NetWare server acting solely as a communications gateway. A server with a run-time version of NetWare for SAA is an example. Thus any potential performance impact on the other NetWare servers is avoided.

Appendix A. Sample NetWare APPC Configuration Definitions

This appendix lists sample VTAM definitions that are required to establish an APPC connection between a NetWare ADSM client and an ADSM/MVS server. The ADSM server's application major node, logmode, and VTAM workstation definitions are covered.

Note: The items highlighted and marked with reference keys, **X**, are values in the NetWare for SAA configuration discussed in 3.4.2.2, "Creating a Service Profile" on page 61 and 3.4.2.3, "Creating a Side Information File" on page 64.

A.1 VTAM Start List

The VTAM start list on an MVS system is usually in SYS1.VTAMLST in a member called ATCSTRxx:

```
SSCPID=06,NOPROMPT,
CONFIG=00,LIST=00,MAXSUBA=31,SUPP=NOSUP,
HOSTSA=6,
SSCPNAME=HOST06,
NETID=USIBMSC,                                          1
HOSTPU=HOST06PU,
CRPLBUF=(208,,15,,1,16),
IOBUF=(100,384,19,,1,20),
LFBUF=(104,,0,,1,1),
LPBUF=(64,,0,,1,1),
SFBUF=(163,,0,,1,1),
WPBUF=(78,,0,,1,1)
```

A.2 VTAM Application Major Node

The following is the VTAM application major node definition for the ADSM server on MVS.

```
ADSM      VBUILD TYPE=APPL
*
*--------------------------------------------------------------------*
*   ADSMLU DEFINITION FOR ADSM                                       *
*                                                                    *
*   ADSM 1.1.0                                                       *
*--------------------------------------------------------------------*
*
SCADSMLU  APPL  ACBNAME=SCADSMLU,                       16
               APPC=YES,
               PARSESS=YES,
               SRBEXIT=YES,
               DMINWNL=3,
               DMINWNR=3,
               DSESLIM=6,
               AUTOSES=3,
```

```
DRESPL=NALLOW,
DDRAINL=NALLOW,
EAS=509,
DLOGMOD=APPCMODE,                          22
MODETAB=MTAPPC,
SECACPT=ALREADYV,
VERIFY=NONE,
VPACING=8
```

A.3 VTAM Logmode Table

Below is the JCL for the job to create the VTAM logmode entry used for ADSM. It creates a logmode entry called APPCMODE in the MTAPPC logmode table, 31 .

```
//TIM1     JOB (999,POK),'MTAPPC',CLASS=A,REGION=4096K,
//    MSGCLASS=T,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//STEP1 EXEC ASMHCL,PARM.L='LIST,REUS'
//C.SYSLIB DD  DSN=SYS1.SISTMAC1,DISP=SHR
//         DD  DSN=SYS1.MACLIB,DISP=SHR
//C.SYSIN  DD  *
MTAPPC   MODETAB                                          31
*************************************************************************
*        LOGMODE TABLE ENTRY FOR APPC CONNECTION.                       *
*************************************************************************
APPCMODE MODEENT LOGMODE=APPCMODE,                        22
               FMPROF=X'13',
               TSPROF=X'07',
               PRIPROT=X'B0',
               SECPROT=X'B0',
               COMPROT=X'D0B1',
               RUSIZES=X'8989',                           26 & 27
               PSERVIC=X'060200000000000000000300'
END      MODEEND
         END
/*
//L.SYSLMOD  DD  DSN=SYS1.LOCAL.VTAMLIB(MTAPPC),DISP=SHR     31
```

The RUSIZES statement, 26 and 27 , defines the maximum inbound and outbound RU size that can be negotiated for use with this logmode. The two bytes, 8989, are the maximum inbound and outbound RU sizes expressed as 8 multiplied by 2 to the power of 9 (8 x 2•) for both inbound and outbound sizes. This equates to inbound and outbound RU sizes of 4096 bytes.

A.4 VTAM Workstation Definitions

The sections that follow include sample VTAM definitions for NetWare for SAA servers attached by 3745 and 3174 gateways.

A.4.1 VTAM 3745 Switched Major Node Definitions

Below is a 3745 switched major node definition for an example of one working configuration. This set of definitions could also be tailored for a 3172 gateway.

```
SWSJ001  VBUILD  TYPE=SWNET,
                 MAXGRP=1,
                 MAXNO=1
*
SJA2024  PU      ADDR=01,                                  12
                 IDBLK=05D,                                7
                 IDNUM=A2024,                              8
                 ANS=CONT,DISCNT=NO,
                 IRETRY=NO,ISTATUS=ACTIVE,
                 MAXDATA=4105,MAXOUT=7,
                 MAXPATH=1,
                 PUTYPE=2,SECNET=NO,
                 MODETAB=MTAPPC,DLOGMOD=APPCMODE,          31 & 22
                 USSTAB=USSRDYN,
                 PACING=8,VPACING=8
*
SJA2024I LU      LOCADDR=0                                 11
*
```

A.4.2 VTAM Cross Domain Definitions

The above workstation definitions were defined on a VM VTAM system, separate from the MVS system where the ADSM server resided. The following cross domain resources were defined to enable the independent LU type 6.2 sessions to connect to the MVS VTAM:

```
          VBUILD TYPE=CDRSC
          NETWORK NETID=USIBMSC                            15
 ********************************************************************
 *                                                                  *
 *   CROSS DOMAIN RESOURCES ON WTSCPOK (LU 6.2)                     *
 *                                                                  *
 ********************************************************************
 *
 SJA2024I CDRSC CDRM=SCG20,                                 11
                ISTATUS=ACTIVE
```

A.4.3 VTAM 3174 Local Node Definitions

Another communications gateway that can be used for APPC connectivity is a 3174. A 3174 is defined in VTAM differently from a 3745. Below is an example of a 3174 local node definition for a 3174 supporting independent LU type 6.2.

This example differs in some parameter selections from the 3745 definition. It also has different PU and LU names for ADSM's APPC sessions. The PU name

of PU3174 is used rather than SJA2024. The LU name of 3174ILU1 is used rather than SJA2024I.

```
DSM3174  VBUILD  TYPE=LOCAL,
*
PU3174   PU    CUADDR=2A0,                          12
               PUTYPE=2,
               XID=YES,
               DISCNT=NO,
               ISTATUS=ACTIVE,
               MAXBFRU=9,
               PACING=8,
               VPACING=8,
               SECNET=NO,
               USSTAB=USSTB7E,
               MODETAB=MTAPPC,                      31
               DLOGMOD=APPCMODE                     22
*
* INDEPENDENT LUs
3174ILU1 LU    LOCADDR=0                            11
*
* DEPENDENT LUs
```

Appendix B. Sample NetWare Server Configuration Files

This appendix contains sample NetWare server STARTUP.NCF and AUTOEXEC.NCF files for the three connectivity methods discussed in this book:

- IPX/SPX
- TCP/IP
- APPC.

These files were used for the configuration of the various connectivity examples shown throughout this book.

B.1 IPX/SPX Connectivity

The following parameters were used with ADSM and the IPX/SPX connectivity protocol:

B.1.1 STARTUP.NCF

```
LOAD PS2ESDI SLOT=8
SET MINIMUM PACKET RECEIVE BUFFERS=500
```

B.1.2 AUTOEXEC.NCF

```
FILE SERVER NAME ITSC-NW2
IPX INTERNAL NET FEAD
LOAD TOKEN
BIND IPX TO TOKEN NET=00000002
MOUNT ALL
LOAD MONITOR
LOAD TSA311
LOAD REMOTE TIM
LOAD RSPX
SEARCH ADD SYS:ADSM
LOAD ROUTE
```

B.2 TCP/IP Connectivity

The following parameters were used with ADSM and the TCP/IP connectivity protocol:

B.2.1 STARTUP.NCF

```
LOAD PS2ESDI SLOT=8
SET MINIMUM PACKET RECEIVE BUFFERS=500
SET MAXIMUM PHYSICAL RECEIVE PACKET SIZE=2040
```

B.2.2 AUTOEXEC.NCF

```
FILE SERVER NAME ITSC-NW1
IPX INTERNAL NET CAFE
LOAD C:\MAINT\TOKEN NAME=IPX-TOKEN
BIND IPX TO OPX-TOKEN NET=00000002
LOAD ROUTE BOARD=1
LOAD MONITOR
SEARCH ADD SYS:/ADSM
LOAD TSA311
LOAD REMOTE TIM
LOAD RSPX
LOAD C:\MAINT\TOKEN FRAME=TOKEN-RING_SNAP NAME=TOKEN-TCP
NODE=520471BC
LOAD TCPIP FORWARD=YES
BIND IP TO TOKEN-TCP ADDR=9.113.36.22 GATE=9.113.36.254
MASK=255.255.255.0
```

B.3 APPC Connectivity

The following parameters were used with ADSM and the APPC connectivity protocol:

B.3.1 STARTUP.NCF

```
LOAD PS2SCSI SLOT=7
SET MINIMUM PACKET RECEIVE BUFFERS=500
SET MAXIMUM PHYSICAL RECEIVE PACKET SIZE = 4200
```

B.3.2 AUTOEXEC.NCF

```
FILE SERVER NAME ITSC-NW3
IPX INTERNAL NET DEAF
LOAD CLIB
LOAD SPXS
LOAD LSLENH
LOAD SPXFIX2
LOAD BTRIEVE -U=1 -P=4096 -F=20 -H=60 -L=20 -C
LOAD TOKEN FRAME=TOKEN-RING SLOT=5 LS=24 SAPS=3 NODE=4011
NAME=TOKENR
BIND IPX TO TOKENR NET=2
LOAD ROUTE BOARD=1
LOAD MONITOR
```

```
LOAD TSA311
MOUNT ALL
LOAD COMMEXEC
LOAD REMOTE TIM
LOAD RSPX
SEARCH ADD SYS:\ADSM
```

Appendix C. Sample NetWare PBTRACE

This appendix contains a sample NetWare for SAA PBTRACE of a working PU type 2.1 service profile and LU type 6.2 side information file. It was created using PBTRACE (see 3.4.3.1, "NetWare PBTRACE" on page 70).

The service profile and side information file used are those discussed in 3.4.2, "Configuration" on page 60. The VTAM definitions are contained in Appendix A, "Sample NetWare APPC Configuration Definitions" on page 107.

Comments in italics provide a brief explanation of the trace. To fully understand a PBTRACE trace you will need to refer to *Novell NetWare Software Developer's Kit, APPC Technical Reference*, 100-001101.

```
NetWare LU6.2    07/12 1994     17:15:37.00
Copyright (C) 1992, Novell, Inc. All Rights Reserved.


API req            00A1452C     17:15:50.00
     00000000 00000000 00000000 1A000000         <................>
     B01BA600 00000000 00000800 3847A100         <..w.............>
     3847A100 00000000 00000000 00000000         <................>
     00000000 00000000 00000000 00000000         <................>
     00000000 00000000 00000000 00000000         <................>
     00000000 00000000 00000000 00000000         <................>

API ret            00A1452C     17:15:50.00
     00000000 00000000 2C45A100 1A000000         <................>
     B01BA600 00000000 00000800 3847A100         <..w.............>
     3847A100 00000000 00000000 00000000         <................>
     00000000 00000000 00000000 00000000         <................>
     00000000 00000000 00000000 00000000         <................>
     00000000 00000000 00000000 00000000         <................>
```

This request and return pair of trace items is for the APPC verb CONVERT. This fact can be determined by the operation code X'1A00' at 12 bytes into each of the trace blocks. The CONVERT verb converts ASCII to EBCDIC. You will see many CONVERT trace pairs in this trace.

```
API req            009FC2A4     17:16:10.00
     00000000 00000000 00000000 20000000         <................>
     B01BA600 00000000 00000000 E4E2C9C2         <..w.........USIB>
     D4E2C340 E2D1C1F2 F0F2F440 00000000         <MSC SJA2024 ....>
     00000000 FFFFFFFF 00000000 00000000         <................>
     00000000 00000000 00000000 00000000         <................>
     00000000 00000000 00000000 00000000         <................>

API ret            009FC2A4     17:16:10.00
     00000000 00000000 A4C29F00 20000000         <........uB......>
     B4C19F00 00000000 04030610 E4E2C9C2         <.A.........USIB>
     D4E2C340 E2D1C1F2 F0F2F440 00000000         <MSC SJA2024 ....>
```

```
     00000000 FFFFFFFF 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
```

This request and return pair of trace items is for the APPC verb ATTACH_PU. This fact can be determined by the operation code X'2000' at 12 bytes into each of the trace blocks. The ATTACH_PU verb requests that a local PU be defined for the APPC session.

The NET_NAME can be found at byte 28 into the block, and the PU_NAME can be found at byte 36 into the block. A return code is available in the return block at byte 20 into the block.

```
API req            009FBFAC     17:16:10.00
     00000000 00000000 00000000 21000000        <................>
     B01BA600 00000000 4600E2D1 C1F2F0F2        <..w.......SJA202>
     F4C90000 00000000 0000000A 00000000        <4I..............>
     00000000 FFFFFFFF 00000000 FFFFFFFF        <................>
     FFFF0000 00003C00 3C002A00 E2C3C1C4        <............SCAD>
     E2D4D3E4 000A0000 4954524E 20202020        <SMLU............>
     00064000 08210200 00000000 00000000        <.. .............>
     00001000 1000C1D7 D7C3D4D6 C4C50010        <......APPCMODE..>
     00010806 00000000 00000000 00000000        <................>

API ret            009FBFAC     17:16:10.00
     00000000 00000000 ACBF9F00 21000000        <................>
     DC9BA100 00000000 4600E2D1 C1F2F0F2        <..........SJA202>
     F4C90200 00000000 0000000A 00000000        <4I..............>
     00000000 FFFFFFFF 00000000 FFFFFFFF        <................>
     FFFF0000 00003C00 3C002A00 E2C3C1C4        <............SCAD>
     E2D4D3E4 000A0000 4954524E 20202020        <SMLU............>
     00064000 08210200 00000000 00000000        <.. .............>
     00001000 1000C1D7 D7C3D4D6 C4C50010        <......APPCMODE..>
     00010806 00000000 00000000 00000000        <................>
```

This request and return pair of trace items is for the APPC verb ATTACH_LU. This fact can be determined by the operation code X'2100' at 12 bytes into each of the trace blocks. The ATTACH_LU verb requests that a local LU be defined for the APPC session.

The LU_NAME can be found at byte 26 into the block, and the PARTNER_LU_NAME can be found at byte 76 into the block. The MODE_NAME can be found at byte 118 into the block. Again, a return code is available in the return block at byte 20 into the block.

```
API req            00A10DA4     17:16:10.00
     00000000 00000000 00000000 2B000000        <................>
     B01BA600 00000000 4954524E 20202020        <..w.............>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
```

```
     00000000 00000000 00000000 00000000        <................>

API ret            00A10DA4     17:16:10.00
     00000000 00000000 A40DA100 2B000000        <........u.......>
     540DA100 00000000 4954524E 20202020        <................>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>

API req            00A72818     17:16:10.00
     00000000 00000000 00000000 15000000        <................>
     B01BA600 00000000 00000200 00000000        <..w.............>
     00000000 00000000 0000E2C3 C1C4E2D4        <..........SCADSM>
     D3E4C1D7 D7C3D4D6 C4C5020A 00000800        <LUAPPCMODE......>
     04000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>

802.2LINK CONNECTED

<=S==    #:005E TH: 2F0001020002 RH: 6B8000
 RU: 31001307 B0B050B1 01008585 80010602        <..........ee....>
     00000000 00000000 23000010 E4E2C9C2        <............USIB>
     D4E2C34B E2D1C1F2 F0F2F4C9 27000902        <MSC.SJA2024I....>
     E2D5C1E2 E5C3D4C7 09030023 F327D2FA        <SNASVCMG....3.K.>
     45731104 E4E2C9C2 D4E2C34B E2D1C1F2        <....USIBMSC.SJA2>
     F0F2F4C9 0008E2C3 C1C4E2D4 D3E4            <024I..SCADSMLU..>
```

At this point in the trace the link has been established, and data begins to flow from the NetWare client to the APPC server. The transmission header (TH), request header (RH), and the request unit (RU) or actual data are displayed in the trace. An SNA technical specialist will be needed to determine whether these fields contain the desired information.

```
==R=>    #:0046 TH: 2F0002010002 RH: EB8000
 RU: 31001307 B0B050B1 01028585 82010602        <..........eeb...>
     00000000 00000012 23000000 27000902        <................>
     E2D5C1E2 E5C3D4C7 0903F023 F327D2FA        <SNASVCMG..0.3.K.>
     45731105 E4E2C9C2 D4E2C34B E2C3C1C4        <....USIBMSC.SCAD>
     E2D4D3E4 0000                              <SMLU............>

<=S==    #:0028 TH: 2E0001020001 RH: 0B9120
 RU: 0F0502FF 0003D000 000206F1 00000000        <...........1....>
     19121002 00000001 000A0000 00000008        <................>
     C1D7D7C3 D4D6C4C5                          <APPCMODE........>

==R=>    #:0000 TH: 2E0002010000 RH: 830100
 RU:                                            <................>

==R=>    #:0019 TH: 2E0002010001 RH: 039101
 RU: 00191210 0A040000 01000600 00000300        <................>
     08C1D7D7 C3D4D6C4 C5                       <.APPCMODE.......>

API ret            00A72818     17:16:13.00
     00000000 00000000 1828A700 15000000        <..........x.....>
```

```
     54ECA800 00000000 00010200 00000000        <..y............>
     00000000 00000000 0000E2C3 C1C4E2D4        <.........SCADSM>
     D3E4C1D7 D7C3D4D6 C4C5020A 00000800        <LUAPPCMODE......>
     04000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>

API req            009B59C0     17:16:13.00
     00000000 00000000 00000000 1A000000        <................>
     B01BA600 00000000 01000800 C8C39F00        <..w.........HC..>
     C8C39F00 00000000 00000000 00000000        <HC..............>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>

API ret            009B59C0     17:16:13.00
     00000000 00000000 C0599B00 1A000000        <................>
     B01BA600 00000000 01000800 C8C39F00        <..w.........HC..>
     C8C39F00 00000000 00000000 00000000        <HC..............>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>

API req            009B59C0     17:16:13.00
     00000000 00000000 00000000 1A000000        <................>
     B01BA600 00000000 01000800 D4C39F00        <..w.........MC..>
     D4C39F00 00000000 00000000 00000000        <MC..............>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>

API ret            009B59C0     17:16:13.00
     00000000 00000000 C0599B00 1A000000        <................>
     B01BA600 00000000 01000800 D4C39F00        <..w.........MC..>
     D4C39F00 00000000 00000000 00000000        <MC..............>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
     00000000 00000000 00000000 00000000        <................>
```

Appendix D. Sample NetWare CSSTATUS Trace

This appendix contains a sample NetWare for SAA CSSTATUS trace of a working PU type 2.1 service profile and LU type 6.2 side information file. It was created using CSSTATUS and formatted with the TRFORMAT utility (see 3.4.3.2, "NetWare CSSTATUS" on page 71).

The service profile and side information file used are those discussed in 3.4.2, "Configuration" on page 60. The VTAM definitions are contained in Appendix A, "Sample NetWare APPC Configuration Definitions" on page 107.

This trace differs from the PBTRACE in that it does not trace all APPC API requests. Comments in italics provide a brief explanation of the trace. A full explanation is beyond the scope of this book.

```
◄ Copyright (C) 1988, 1990 Novell, Inc.◄ Release 2.0
◄ Date: 07/01/94 19:48:12◄

RET     TIME:00:00:00.00  VPL = ATTACH_PU
     00000000 00000000 00000000 0819B900 23200000 00000000 00000003 00010000
     43000000 00000000 7818B900 E2D1C1F2 F0F2F440 E4E2C9C2 D4E2C340 FFFFFFFF
     000000

REQ     TIME:00:00:00.00  VPL = ATTACH_PU
     00000000 00000000 00000000 0819B900 23200000 00000000 00000003 00010000
     43000000 00000000 7818B900 E2D1C1F2 F0F2F440 E4E2C9C2 D4E2C340 FFFFFFFF
     061000
```

The trace is made up of request and reply pairs. The APPC request that initiated the above pair is an ATTACH_PU. The PU_NAME and NET_NAME can be found in the data displayed.

```
RET     TIME:00:00:00.00  VPL = ATTACH_LU
     00000000 00000000 00000000 0819B900 22200000 00000000 00000002 00010000
     4C000000 00000000 2016B900 0100E2D1 C1F2F0F2 F4C9000A 00000000 FFFFFFFF
     FFFFFFFF FF01FF00 00000000

RET     TIME:00:00:00.00  VPL = ATTACH_LU
     00000000 00000000 00000000 0819B900 22200000 00000000 00000000 00000000
     54000000 00000000 2016B900 0200E2C3 C1C4E2D4 D3E4000A 00004954 524E2020
     20200006 40000821 02000000 00000000 00000000

RET     TIME:00:00:00.00  VPL = ATTACH_LU
     00000000 00000000 00000000 0819B900 22200000 00000000 00000001 00000000
     3C000000 00000000 2016B900 0300C1D7 D7C3D4D6 C4C50010 00010000

REQ     TIME:00:00:00.05  VPL = ATTACH_LU
     F003B900 00000000 00000000 0819B900 22200000 00000000 00000003 00010000
     4C000000 00000000 2016B900 0100E2D1 C1F2F0F2 F4C9000A 00000000 FFFFFFFF
     FFFFFFFF FF01FF04 00000000
```

```
RET     TIME:00:00:00.05  VPL = ACTIVATE_DLC
     00000000 00000000 00000000 0819B900 21200000 00000000 00000003 00010000
     35000000 00000000 9055BC00 4954524E 20202020 00

REQ     TIME:00:00:00.05  VPL = ACTIVATE_DLC
     00000000 00000000 00000000 0819B900 21200000 00000000 00000003 00010000
     35000000 00000000 9055BC00 4954524E 20202020 00

RET     TIME:00:00:00.05  VPL = CNOS
     00000000 00000000 00000000 0819B900 25200000 00000000 00000003 00060000
     43000000 00000000 C833C300 0402040A E2C3C1C4 E2D4D3E4 C1D7D7C3 D4D6C4C5
     00000A
```

The following entries in the trace display the control flows during the establishment of the APPC session:

```
<=I=>   TIME:00:00:00.05    #:0016  TH: 2C0000040004 RH: 0B8000
 SEG=ONLY   DAF:00  OAF:04  SEQ:0004
 RU=REQ,DR  RC=FMD,RQCPID  CHAIN=ONLY
 RU: 81062700 0108E2C3 C1C4E2D4 D3E4E2D5 C1E2E5C3 D4C7

<=I=>   TIME:00:00:00.05    #:0005  TH: 2C0000000000 RH: 0B8000
 SEG=ONLY   DAF:00  OAF:00  SEQ:0000
 RU=REQ,DR  RC=NSH,CONTACT  CHAIN=ONLY
 RU: 01020104 00

<=I=>   TIME:00:00:00.05    #:0003  TH: 2C0000000000 RH: 8B8000
 SEG=ONLY   DAF:00  OAF:00  SEQ:0000
 RU=RSP+ RC=NSH,CONTACT  CHAIN=ONLY
 RU: 010201

<=I=>   TIME:00:00:01.05    #:0005  TH: 2C0000000007 RH: 0B8000
 SEG=ONLY   DAF:00  OAF:00  SEQ:0007
 RU=REQ,DR  RC=NSH,XXXXX  CHAIN=ONLY
 RU: 01028004 00

<=I=>   TIME:00:00:01.05    #:0003  TH: 2C0000000007 RH: 8B8000
 SEG=ONLY   DAF:00  OAF:00  SEQ:0007
 RU=RSP+ RC=NSH,XXXXX  CHAIN=ONLY
 RU: 010280

<=I=>   TIME:00:00:01.05    #:001b  TH: 2C0004000004 RH: 8B8000
 SEG=ONLY   DAF:04  OAF:00  SEQ:0004
 RU=RSP+ RC=FMD,RQCPID  CHAIN=ONLY
 RU: 81062700 0108E2C3 C1C4E2D4 D3E4E2D5 C1E2E5C3 D4C70404 010102

<=I=>   TIME:00:00:01.05    #:0022  TH: 2C0000040004 RH: 0B8000
 SEG=ONLY   DAF:00  OAF:04  SEQ:0004
 RU=REQ,DR  RC=NSH,INIT-SELF  CHAIN=ONLY
 RU: 81068110 C2C20000 E2D5C1E2 E5C3D4C7 F308E2C3 C1C4E2D4 D3E40000 00040404
     0101

<=I=>   TIME:00:00:01.05    #:0008  TH: 2C0004000000 RH: 0B0000
 SEG=ONLY   DAF:04  OAF:00  SEQ:0000
 RU=REQ,NR  RC=NSH,CINIT  CHAIN=ONLY
```

```
 RU: 81060104 04010100

<=I=>   TIME:00:00:01.05    #:0003  TH: 2C0004000004 RH: 8B8000
 SEG=ONLY   DAF:04  OAF:00  SEQ:0004
 RU=RSP+ RC=NSH,INIT-SELF  CHAIN=ONLY
 RU: 810681
```

The following entries in the trace display the data flows and their responses after the establishment of the APPC session:

```
<=S==   TIME:00:00:01.05    #:005e  TH: 2F0001040004 RH: 6B8000
 SEG=ONLY   EFI=ON  DAF:01  OAF:04  SEQ:0004
 RU=REQ,DR  RC=SC,BIND  CHAIN=ONLY
 RU: 31001307 B0B050B1 01008585 80010602 00000000 00000000 23000010 E4E2C9C2
     D4E2C34B E2D1C1F2 F0F2F4C9 27000902 E2D5C1E2 E5C3D4C7 090300B7 985B1AFA
     947C1104 E4E2C9C2 D4E2C34B E2D1C1F2 F0F2F4C9 0008E2C3 C1C4E2D4 D3E4

==R=>   TIME:00:00:02.10    #:0046  TH: 2F0004010004 RH: EB8000
 SEG=ONLY   EFI=ON  DAF:04  OAF:01  SEQ:0004
 RU=RSP+ RC=SC,BIND  CHAIN=ONLY
 RU: 31001307 B0B050B1 01028585 82010602 00000000 00000012 23000000 27000902
     E2D5C1E2 E5C3D4C7 0903F0B7 985B1AFA 947C1105 E4E2C9C2 D4E2C34B E2C3C1C4
     E2D4D3E4 0000

<=I=>   TIME:00:00:02.10    #:0008  TH: 2C000004000E RH: 0B0000
 SEG=ONLY   DAF:00  OAF:04  SEQ:000E
 RU=REQ,NR  RC=NSH,SESSST  CHAIN=ONLY
 RU: 81068604 04010101

<=S==   TIME:00:00:02.10    #:0028  TH: 2E0001040001 RH: 0B9120
 SEG=ONLY   DAF:01  OAF:04  SEQ:0001
 RU=REQ,EX  RC=FMD,FMH  CHAIN=ONLY  CDI=ON  PI=ON
 RU: 0F0502FF 0003D000 000206F1 00000000     <...........1....>
     19121002 00000001 000A0000 00000008     <................>
     C1D7D7C3 D4D6C4C5                       <APPCMODE        >

==R=>   TIME:00:00:02.20    #:0000  TH: 2E0004010000 RH: 830100
 SEG=ONLY   DAF:04  OAF:01  SEQ:0000
 RU=RSP+ RC=FMD,DATA  CHAIN=ONLY  PI=ON

==R=>   TIME:00:00:02.25    #:0019  TH: 2E0004010001 RH: 039101
 SEG=ONLY   DAF:04  OAF:01  SEQ:0001
 RU=REQ,EX  RC=FMD,DATA  CHAIN=ONLY  BRACKET=CEB  PI=ON
 RU: 00191210 0A040000 01000600 00000300     <................>
     08C1D7D7 C3D4D6C4 C5                    <.APPCMODE       >
```

Index

A

administrative client 8, 9, 21
administrator
  See storage administrator
ADSM
  console 32, 75
  releases 21, 43
  server
    See server
  startup 45, 74—76, 96
advanced program-to-program communications
  See APPC
alternate path
  See cross-directory
API 10, 21, 35, 39
APPC
  ADSM support 6, 21, 56
  associated VTAM parameters 56—58
  configuring for ADSM 60, 61, 68, 69, 98, 107—109, 112
  creating a service profile 61—63
  creating a side information file 64—68
  definition 6, 34, 56
  performance 72—74
  problem determination 69—71, 115, 119
  system requirements 58, 59
APPCRESOURCE 67
application client 10, 21
application programming interface
  See API
archive
  ARCHIVE 85
  client options 20
  definition 3
  delete 86
  description parameter 85, 86
  functions 11, 12, 13, 99
  scheduling 87
  usage 85, 86
attributes 31, 35, 37, 78—81
authentication 9
authorization 9, 101
AUTOEXEC.NCF file 24, 25, 44, 45, 52, 53, 60, 61, 63, 93, 95, 98, 111, 112
availability management 2, 3, 34, 35

B

backup
  bindery 82—83
  client options 20
  definition 2, 3
  functions 7, 10, 11, 13, 36, 37, 99
  incremental 10, 11, 17, 18, 20, 46, 77, 78, 82, 84, 99
backup *(continued)*
  NDS 84, 85
  query 82
  scheduling 87
  selective 10, 78, 79, 82—84
  versions 2, 3, 10, 11, 17, 81, 82
backup/archive client 7
BIND 49, 52, 60, 73
bindery 20, 22, 29, 35, 37, 46, 82—84, 95, 97, 99
BINDFIX utility 84
binding 56, 57
block ID 62, 63, 65
BTRIEVE 60
buffering 50, 51, 54, 55, 57, 61, 72, 73

C

C interface library 60
central scheduling
  See scheduling
CLI 8, 21
client
  definition 1, 2, 4
  functions 7
  installing 43, 44
  multiple clients 101—103
  multiple clients and different servers 92, 93
  multiple clients and single server 91, 92
  options 19, 20
  registration 9
  single client and multiple servers 94—100
client options file
  See DSM.OPT file
client/server 1, 2, 4, 9, 73
CNOS 68
collision 11
command line interface
  See CLI
COMMEXEC 60
COMMMETHOD 50, 53, 68
common names 38
common programming interface for communications
  See CPIC
communications gateway
  See gateway
communications protocol
  ADSM client/server 6, 21
  APPC
    See APPC
  client options 20
  IPX/SPX
    See IPX/SPX
  PWSCS
    See PWSCS
  TCP/IP
    See TCP/IP

compression 12, 20, 38, 39, 99, 100
connectivity
See communications protocol
console
See ADSM
See NetWare
container objects 37
copy group
archive 16, 17, 85
backup 16, 17
definition 15—17
DESTINATION 17
FREQUENCY 17
MODE 18
RETEXTRA 17
RETONLY 17
SERIALIZATION 18
VERDELETED 17
VEREXISTS 17
CPIC 64
CPICBUFFERSIZE 72, 73
CPNAME 62, 63, 65
cross-client 101
restore 101—104
cross-directory
restore 11, 81
retrieve 86
cross-domain 66, 109
cross-platform
restore 13
retrieve 13
cross-user
restore 13
retrieve 13
CSCON utility 61, 62, 64, 65, 68
CSLOAD 63
CSSTATUS trace 69, 71, 119

D

data compression
See compression
data security
See security
DELETE 86
DESCRIPTION 85, 86
DESTINATION 17
directory
NetWare
See NetWare
standard file system 2, 10, 27—30, 46, 78—81, 86, 93, 98
disaster recovery 3, 92
disk duplexing
See disk mirroring
disk mirroring 34
DOMAIN 20, 46, 78, 82, 84, 95, 96, 99
DOWN 45
DSM.OPT file
COMMMETHOD 53, 68, 74
communications options 74, 91
COMPRESSION 99
configuring ADSM 45—47
CPICBUFFERSIZE 72, 73
definition 19, 20
directory 92, 93
DOMAIN 20, 46, 78, 82, 84, 95, 96, 99
EXCLUDE 10, 11, 20, 46, 47, 78, 79, 96
INCLUDE 10, 11, 16, 20, 46, 47, 78, 79, 96
MAXCMDRETRIES 89
NODENAME 45
NWPWFILE 47, 75, 97
NWUSER 47
QUERYSCHEDPERIOD 89
REPLACE 80, 81
RETRYPERIOD 89
SCHEDLOGNAME 89
SCHEDMODE 89
scheduling options 89
SUBDIR 80, 81
SYMBOLICDESTINATION 65, 68
TCPBUFFSIZE 55
TCPCLIENTADDRESS 90
TCPCLIENTPORT 90
TCPPORT 53
TCPSERVERADDRESS 53
TCPWINDOWSIZE 55
DSM.SMP file 45
DSMC 44, 45, 75—77, 87, 91—93, 96
dynamic LU
See LU

E

encryption 47
EXCLUDE 10, 11, 20, 46, 47, 78, 79, 96
EXIT 45
export 3

F

file attributes
See attributes
file space
See FILESPACE
file support 12, 35
FILESPACE 98, 99
FREQUENCY 17

G

gateway 22, 52, 58, 59, 61, 66, 94, 99, 108, 109
graphical user interface
See GUI
GUI 8, 21

H

hot fix 34
hot standby
  See server, mirroring

I

IDBLK
  See block ID
IDNUM
  See PU ID
import 3
INCLUDE 10, 11, 16, 20, 46, 47, 78, 79, 96
INCREMENTAL 77, 84
incremental backup 10, 11, 17, 18, 20, 46, 77, 78
Internet protocol
  See TCP/IP
internetwork packet exchange protocol
  See IPX/SPX
IPX/SPX
  ADSM support 6, 21, 48
  configuring for ADSM 48—50, 111
  definition 6, 34
  performance 50, 51
  problem determination 50
  remote NetWare servers 99, 102, 103
  system requirements 48, 94
  use with APPC 60
IPXBUFFERSIZE 50
IPXSERVERADDRESS 50
IPXSOCKET 50

L

LAN 25, 34, 49, 51, 52, 54, 58, 60, 62, 63, 66, 72, 73, 94
leaf objects 37, 38
LINK DRIVER 49
link stations 60
LOAD 43, 44, 52, 60, 68, 70, 71, 75, 77, 79, 87, 91—93, 95, 96
local area network
  See LAN
logical unit
  See LU
logmode table 57, 67, 68, 73, 108
LU
  definition 56
  dependent 56
  dynamic LU 65, 68
  independent 56, 58, 59, 62, 65
  LU type 6.2 56, 58, 59, 62, 66—69, 107, 108, 109
  partner LU 57, 64, 66—68

M

management class 13, 15, 16, 79, 81, 85
MAP 27, 28, 32
MAXCMDRETRIES 89
migration 38, 39
MODE 18

N

NCP 59, 74
NDS
  NetWare version 4 2, 22, 37—39, 46, 84, 85, 99
  partitioned 38
  replication 38, 84, 85
NET.CFG file 49, 51
NetBIOS and IPX/SPX Support/6000 48
NetWare
  administrator 31, 44, 76
  AUTOEXEC.NCF file 24, 25, 44, 52, 53, 60, 61, 63, 93, 95, 98, 111, 112
  bindery 20, 22, 29, 35, 37, 46, 82—84, 95, 97, 99
  console 32, 33, 71, 91
  directory 2, 20, 22, 28, 37—39, 46, 84, 85, 99
  disk partition 26, 27
  file formats 28
  file path identification 27, 28
  file system 26—28
  files 38, 39
  functions 23
  name spaces 28
  operating system 23, 24, 26
  remote servers 36, 103, 104
  server
    See server
  shutdown 45
  startup 23, 24, 29, 44
  STARTUP.NCF file
    See STARTUP.NCF file
  SYS: volume 24, 27, 28, 43, 46, 100
  unique ADSM functions 22
  unsupported ADSM functions 21
  use of DOS 23
  version 4 differences 37—40
  versions 21, 23, 37, 40, 43, 45, 59, 95
  volumes 27, 46
  workstation support 23
NetWare directory services
  See NDS
NetWare for SAA
  APPC parameters 57, 60, 73
  definition 40
  functions 40
  packaging 40, 41
  problem determination 69—71, 115, 119
  required usage 21, 23, 34, 59, 98
  run-time version 41, 98
  service profile parameters 61—63
  side information file parameters 64—68
  unload profiles 69
NetWare for UNIX AIX 48

NetWare loadable modules
See NLM
network address 49, 50
Network Control Program
See NCP
network ID 57, 62, 66
NLM 24—26, 32
node name 2, 9, 45, 47, 62, 96
NODENAME 45
NWPWFILE 47, 75, 97
NWUSER 47

O

objects
See container objects
See leaf objects
organization units 38, 84, 85

P

pacing 67, 68, 72, 73
partner LU
See LU
password 9, 47, 75, 76, 97
PBTRACE 69—71, 115
performance 56
ADSM server 12
APPC 72—74
client 12, 99
compression 99, 100
IPX/SPX 50, 51
network 99, 103
TCP/IP 54—56
physical unit
See PU
PING 53, 54
policy domain 13, 14
policy management 3, 10, 11, 13—17
policy set 13, 15
problem determination
APPC 69, 70, 71
IPX/SPX 50
TCP/IP 53, 54
PU 57—63, 65, 66, 108, 109
PU ID 62, 63, 65
PU type 2.0
See PU
PU type 2.1
See PU
PWSCS 6, 21, 43

Q

QUERY 98, 101
QUERY BACKUP 82
QUERYSCHEDPERIOD 89
QUIT 76, 92

R

RACF 9
RCONSOLE 33, 44
recall 3
recovery
data 3
disaster 3, 92
disk 3, 100—103
server 100
registration
closed 9
open 9
REPLACE 80, 81
requester 1, 2, 12, 32, 33, 44, 48, 49, 99
response unit
See RU
restore
bindery 82—84
client options 20
collision 11
cross-client 101—104
cross-directory 11, 81
cross-platform 13
cross-user 13
definition 3
functions 7, 10, 11, 13, 36, 37
NDS 84, 85
RESTORE 79—81, 83, 85, 102, 104
usage 79—82
RETEXTRA 17
RETONLY 17
retrieve
by date 12
by description 12
client options 20
cross-directory 86
cross-platform 13
cross-user 13
definition 3
functions 7, 12, 13
RETRIEVE 85, 86
usage 85, 86
RETRYPERIOD 89
rights 30, 31, 33, 35, 37, 44, 61, 77—81, 84
RU 67—69, 72, 73, 108
run-time version
See NetWare for SAA

S

SAP 60, 63
SCHEDLOGNAME 89
SCHEDMODE 89
SCHEDULE 87
scheduling
client options 20
client polling 19, 89
definition 13, 18, 19

scheduling *(continued)*
functions 92
server prompted 19, 89, 90
usage 87—90
SEARCH 26, 44, 93
security
ADSM functions 9
file system 30, 31
NetWare functions 22, 47
NetWare server access 29
SELECTIVE 78, 83
selective backup 10, 78, 79
sequenced packet exchange protocol
See IPX/SPX
SERIALIZATION 18
server
ADSM server 1, 4
AIX/6000 configurations 6, 21
APPC definitions 107—109
definition 1, 2, 4
different servers and multiple clients 92, 93
functions, ADSM 7
mirroring 35
multiple servers and single client 94—100
MVS configurations 6, 21
NetWare server 1
OS/2 configurations 6, 21
recovery 100
remote NetWare servers 36, 103, 104
single server and multiple clients 91, 92
VM configurations 6, 21
service access point
See SAP
service profile 61—63
sessions 62, 66, 67, 93
SET 54, 61, 101, 104
SFT 34, 35
side information file 64—68
SIUTIL utility 64, 68
sliding window 55
SMDE 36
SMDR 39, 44, 45, 94, 95, 102, 103
SME 36, 39
SMS 22, 35, 36, 37, 39, 40, 43, 82, 95, 102, 103
SNA 6, 34, 40
socket number 50
space management 3, 85
STARTUP.NCF file 24, 111, 112
storage administrator 4, 8—10, 13, 14, 19
storage hierarchy 7
storage management data requester
See SMDE
storage management engine
See SME
storage management services
See SMS
storage pools 7, 17
SUBDIR 80, 81
SYMBOLICDESTINATION 65, 68
system fault architecture
See SFT
Systems Network Architecture
See SNA

T

target service agent
See TSA
TCP/IP
ADSM support 6, 21, 51
configuring for ADSM 52, 53, 112
definition 6, 34, 51
performance 54—56
problem determination 53, 54
scheduling 19, 90
system requirements 51
TCPBUFFSIZE 55
TCPCLIENTADDRESS 90
TCPCLIENTPORT 90
TCPPORT 53
TCPSERVERADDRESS 53
TCPWINDOWSIZE 55
terminate and stay resident program
See TSR
token-ring 49, 51, 52, 54, 58, 60, 62, 63, 66, 72, 94
trace 69—71, 115, 119
transaction backout
See TTS
transaction tracking system
See TTS
transmission control program
See TCP/IP
TRFORMAT utility 71, 119
trustee rights
See rights
TSA 37, 39, 44, 45, 94, 95, 102, 103
TSR 39, 40
TTS 35

U

UNLOAD 76, 92
user ID 31, 47, 75, 97
user interface 8, 20, 21, 32, 33

V

VERDELETED 17
VEREXISTS 17
versions
active 10, 81
backup
See backup
inactive 10, 81
vital records 3, 85

VM 67
VTAM 56—59, 61—63, 65—71, 73, 107—109

W

wildcards 10, 12, 96
workstation interface 20, 39, 40

ITSO Technical Bulletin Evaluation RED000

Getting Started with ADSM NetWare Clients

Publication No. GG24-4242-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____

Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

_____

_____

What other topics would you like to see in this Bulletin?

_____

_____

What other Technical Bulletins would you like to see published?

_____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

_____ _____
Name Address

_____ _____
Company or Organization

_____ _____
Phone No.

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 471, Building 070B
5600 COTTLE ROAD
SAN JOSE CA
USA 95193-0001

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.