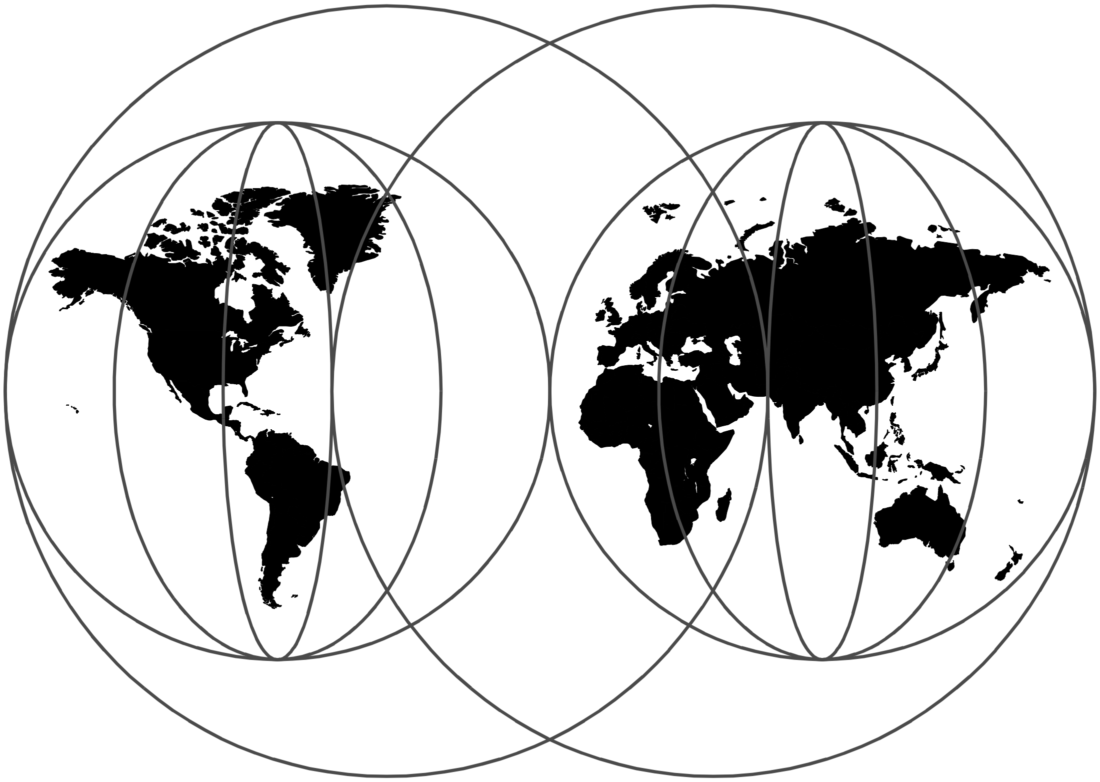
ADSM Version 3 Technical Guide

Tim Mortimer, Claudio Frignani, Lenhle Khoza, Walter Majonica



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-2236-01

International Technical Support Organization

SG24-2236-01

ADSM Version 3 Technical Guide

December 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A., “Special Notices” on page 349.

Second Edition (December 1998)

This edition applies to Version 3 Release 1 of ADSTAR Distributed Storage Manager for AIX, Program Number 5765-C43; Version 3 Release 1 of ADSTAR Distributed Storage Manager for HP-UX, Program Number 5639-D92; Version 3 Release 1 of ADSTAR Distributed Storage Manager for MVS, Program Number 5655-A30; Version 3 Release 1 of ADSTAR Distributed Storage Manager for Sun Solaris, Program Number 5639-D91; Version 3 Release 1 of ADSTAR Distributed Storage Manager for Windows NT, Program Number 5639-C59.

Note

This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. We recommend that you consult the product documentation or follow-on versions of this redbook for more current information.

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

Contents

Tables ix

Preface xi
The Team That Wrote This Redbook xi
Comments Welcome xii

Chapter 1. ADSM Version 3 Overview 1
1.1 Background. 2
1.1.1 ADSM Version 2 2
1.1.2 Emerging Trends. 4
1.1.3 ADSM Version 3 Strategy 6
1.2 ADSM Version 3 Overview 7
1.2.1 New Backup-Archive GUI 7
1.2.2 New Backup-Archive Client Function. 9
1.2.3 Backup-Archive Performance 11
1.2.4 Server and Network Performance 12
1.2.5 Administrative Interfaces 14
1.2.6 Web Backup-Archive Client. 16
1.2.7 Server SQL Interface. 18
1.2.8 Administration Enhancements 19
1.2.9 Server and Client Configuration 21
1.2.10 Server-to-Server Communications. 22
1.2.11 Enterprise Configuration 24
1.2.12 Server-to-Server Virtual Volumes 26
1.2.13 Media Management. 28
1.2.14 Event Logging, Reporting, and Monitoring. 29
1.2.15 ADSMConnect Agents. 31
1.2.16 ADSM Version 3 Summary 32

Chapter 2. Backup-Archive Client 33
2.1 Backup-Archive Client GUI 34
2.1.1 Backup-Archive Client GUI Features 36
2.1.2 Find and File Detail Functions. 39
2.1.3 Estimate Function 41
2.1.4 Graphical Options Editor 42
2.2 New Backup and Restore Functions 44
2.2.1 Directory Backup Support 46
2.2.2 Exclude Directories from Backup. 48
2.2.3 Restore to New Location 50
2.2.4 No Query Restore Protocol 52
2.2.5 Client Restore Processes 54
2.2.6 Point-in-Time Restore 56
2.2.7 Point-in-Time Restore Options 58
2.2.8 Point-In-Time Restore Considerations. 59
2.2.9 Point-in-Time Restore Policy Definitions 61
2.2.10 Tape Mount Prompt. 62
2.3 Client Fault Tolerance. 64
2.3.1 File and Volume Errors 65
2.3.2 Communications Errors 66
2.3.3 Restartable Restores. 68
2.3.4 Restartable Restore Commands 70

2.3.5 Restartable Restore Database Information 73
2.4 Archive and Retrieve Enhancements 74
2.4.1 Archive Descriptions 75
2.4.2 Archive Packages 76
2.4.3 Retrieving Archive Packages 78
2.4.4 Directory Archiving. 79
2.5 Other Client Enhancements. 81
2.5.1 Previous Command Recall. 82
2.5.2 Passwordaccess Generate 84
2.5.3 Enhanced Client File System Support 86
2.5.4 Enhanced Online Help and Documentation 88

Chapter 3. Performance Enhancements. 91
3.1 Server Enhancements 92
3.1.1 Server File Aggregation 93
3.1.2 Server File Aggregation 95
3.1.3 Support for Larger Buffers 98
3.1.4 Other Server Enhancements 100
3.2 Client Enhancements. 101
3.2.1 No Query Restore Protocol 102
3.2.2 Fault Tolerance 103
3.2.3 Enhanced Confirmation Processing 104
3.2.4 Enhanced Control 106

Chapter 4. Administrative Interfaces 109
4.1 Web Administrative Interface. 110
4.1.1 ADSM Enterprise Console 112
4.1.2 Web Administrative User Interface. 114
4.1.3 Web Administrative Implementation. 116
4.1.4 Web Administrative Security 118
4.1.5 Web Session Authentication 119
4.1.6 Secure Sockets Layer (SSL) 120
4.2 Win32 GUI Administrative Client 123
4.2.1 Win32 Administrative User Interface 124
4.2.2 Win32 Administrative Client Function. 125
4.3 Server SQL Interface. 127
4.3.1 Administrative SELECT Command. 128
4.3.2 SQL SELECT Statement Syntax 130
4.3.3 SQL Select Statement Example. 134
4.3.4 Server SQL Processing 135
4.3.5 Additional SQL Commands 136
4.3.6 ODBC Interface 138
4.3.7 ODBC Implementation 139
4.3.8 ODBC Configuration 140

Chapter 5. New Administrative Functions 143
5.1 Server and Client Configuration. 144
5.1.1 Server Options Online 145
5.1.2 Query System Command 147
5.1.3 Client Option Sets 149
5.1.4 Defining Option Sets 150
5.1.5 Sample Option Set. 152
5.1.6 Enhanced Client Information 153
5.2 Scheduling and Security 154

5.2.1 One-Time Scheduled Client Commands .155
5.2.2 Signon Security 157
5.2.3 Password Security 159
5.3 Server Automation 160
5.3.1 Server Scripts 161
5.3.2 Web Administrative Script Interface. 164
5.3.3 Script Syntax. 165
5.3.4 Script Logic 166
5.3.5 Logic Statements 167
5.3.6 Sample Scripts 169
5.3.7 Database and Log Management Overview 171
5.3.8 Database and Log Utilization Triggers. 172
5.3.9 Database and Log Expansion Commands. .173
5.3.10 Synchronous Administrative Commands .175

Chapter 6. Enterprise Administration 177
6.1 Server-to-Server Communications 178
6.1.1 Server-to-Server Overview 179
6.1.2 Server-to-Server Configuration 180
6.1.3 Remote Server Setup 181
6.1.4 Local Server Setup 183
6.1.5 Server Definitions 185
6.1.6 Server Groups. 187
6.1.7 Server Group Configuration. 188
6.2 Administrative Command Routing 190
6.2.1 Command Routing Setup 191
6.2.2 Command Routing Examples 193
6.3 Enterprise Configuration 194
6.3.1 Configuration Manager 195
6.3.2 Configuration Profiles 197
6.3.3 Configuration Profile Commands (1) 199
6.3.4 Configuration Profile Commands (2) 201
6.3.5 Managed Servers 203
6.3.6 Managed Administrators 206
6.3.7 Managed Policy Domains 208
6.3.8 Managed Administrative Command Schedules 210
6.3.9 Managed Server Scripts 211
6.3.10 Managed Client Option Sets 212
6.3.11 Managed Servers and Server Groups 213
6.3.12 Profile Propagation 215
6.3.13 Profile Subscribers 217

Chapter 7. Enterprise Login 219
7.1 Single Administrative Login Overview 220
7.2 Server Single Administrative Login 221
7.2.1 Remote Server Configuration 222
7.2.2 Remote Server Access 224
7.3 Web Backup-Archive Client. 226
7.3.1 Web Client Introduction 227
7.3.2 Client Owner Authority 229
7.3.3 Client Access Authority 232
7.3.4 Web Client Configuration 234
7.3.5 Hyperlink Client Connection 236

7.3.6 Hyperlink Client Authentication 237
7.3.7 Direct Web Client Authentication 239

Chapter 8. Server Device Configuration. 241
8.1 Server-to-Server Virtual Volumes 242
8.1.1 Virtual Volumes Overview 243
8.1.2 Configuration (1) 244
8.1.3 Configuration (2) 246
8.1.4 Configuration (3) 248
8.1.5 Database Backup 250
8.1.6 Primary Storage Pool. 251
8.1.7 Copy Storage Pool. 252
8.1.8 Virtual Volumes 253
8.1.9 Managing Virtual Volumes 256
8.1.10 Virtual Volumes Considerations 259
8.1.11 DRM Virtual Volume Integration. 261
8.1.12 Remote DRM Plan File (1) 264
8.1.13 Remote DRM Plan File (2) 265
8.2 Tape Management. 269
8.2.1 Overflow Storage Pools 270
8.2.2 Defining Overflow Storage Pools 271
8.2.3 Overflow Storage Pool Media Movement 272
8.2.4 Overflow Storage Pool Mount Requests. 274
8.2.5 Overflow Storage Pool Macro Generation 275
8.2.6 Sample Overflow Storage Pool Definition. 277
8.2.7 Sample Overflow Storage Pool Mount Request 278
8.2.8 Migration by Age 279
8.2.9 ADSM Version 2 Mount Limit Management 282
8.2.10 ADSM Version 3 Mount Limit Management 283
8.2.11 Monitoring Mount Limits. 285
8.2.12 Single Drive Reclamation. 287
8.2.13 Single Drive Reclamation Example 289
8.2.14 Single Command Label and Check in 290

Chapter 9. Event Logging, Reporting, and Monitoring 291
9.1 Monitoring Client Events 292
9.1.1 Client Message Candidates 293
9.1.2 Client Event Formatting 295
9.1.3 Client Event Processing. 296
9.1.4 Enabling Client Events for the ADSM Console 297
9.1.5 Storing Client Events in the Activity Log. 298
9.1.6 Querying Client Events in the Activity Log 299
9.2 Event Reporting. 300
9.2.1 Event Logging Commands 302
9.2.2 Sending Events to a File Exit 306
9.2.3 Sending Events to a User Exit 308
9.2.4 Sending Events to NT Eventlog 310
9.2.5 Sending Events to a Remote Server 312
9.2.6 Sending Events to the Tivoli/Enterprise Console 314
9.2.7 Customizing T/EC for ADSM Events 316
9.2.8 Sending Events to NetView for MVS 317
9.3 SNMP Monitoring and Reporting 319
9.3.1 SNMP Terminology 320

9.3.2 SNMP Agent 322
9.3.3 Manager-Agent-Subagent Communication 324
9.3.4 ADSM SNMP Implementation 326
9.3.5 Subagent Processing 327
9.3.6 Monitoring and Traps 328
9.3.7 Server Monitoring 329
9.3.8 SNMP Server Options 331
9.3.9 SNMP Setup 333

Chapter 10. Planning Considerations 335
10.1 Platform Support 336
10.1.1 Version 3 Servers 336
10.1.2 Version 3 UNIX Clients 337
10.1.3 Version 3 PC Clients 338
10.2 Versions 2 and 3 Compatibility 339
10.2.1 Versions 2 and 3 Compatibility 339
10.2.2 Versions 2 and 3 Compatibility... 340
10.2.3 Administrative Client Compatibility 341
10.3 Licensing and Packaging 342
10.3.1 ADSM for UNIX Servers Licensing 342
10.3.2 ADSM for MVS Licensing 344
10.3.3 ADSM for Windows NT Licensing 346

Appendix A. Special Notices 349

Appendix B. Related Publications 351
B.1 International Technical Support Organization Publications 351
ADSM Redbooks 351
B.2 Redbooks on CD-ROMs 352
B.3 ADSM Product Publications 352
B.4 ADSM Online Product Library 353

How to Get ITSO Redbooks 355
How IBM Employees Can Get ITSO Redbooks 355
How Customers Can Get ITSO Redbooks 356
IBM Redbook Order Form 357

Index 359

ITSO Redbook Evaluation 369

Tables

1. Version 3 Server and Client NLS Matrix 89
2. TXNGROUPMAX and TXNBYTELIMIT Values 96
3. Version 3 Maximum Tape Block Sizes. 99

Preface

IBM ADSTAR Distributed Storage Manager (ADSM) is an enterprisewide network storage management solution. ADSM Version 3 introduces new functions such as enhanced user interfaces, improved performance, server-to-server communications, and integration with systems management applications.

The book is structured as an overview followed by detailed product information. The first chapter provides the overview of the new functions. Subsequent chapters provide the detailed descriptions of the new functions. This second edition of the book covers the Enterprise Administration enhancements introduced with ADSM Version 3.1.2.

This redbook is intended for customers, consultants, IBM Business Partners, and IBMers who are familiar with ADSM Version 2 and need to understand what is new in Version 3.

The Team That Wrote This Redbook

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

Tim Mortimer is a Senior Systems Specialist at the International Technical Support Organization-San Jose Center. He writes extensively and teaches IBM classes worldwide on all areas of ADSM and distributed storage management. Before joining the ITSO in 1996, Tim worked for IBM in the United Kingdom as a storage management consultant. He has 20 years of IT experience, including 10 years of storage management experience covering a wide range of mainframe and client/server platforms.

Claudio Frignani is an IT Specialist in Italy. He is ADSM certified and has three years of experience in ADSM across all supported platforms. Claudio’s other areas of expertise include MVS storage management, RAMAC data migrator, RVA, and Snapshot. Before joining the storage group in 1995, Claudio was a RACF and security specialist.

Lenhle Khoza is an IT Specialist in South Africa. She is ADSM certified, with two years experience in ADSM. She has three years of IT experience, including a year of storage management experience on MVS. She holds a BSc degree in computer science from the University of the Western Cape (UWC) in South Africa. Her areas of expertise include MVS storage management and AIX.

Walter Majonica is a Systems Management Specialist in Germany. He has eight years of experience as a storage management consultant and five years of experience with ADSM on a wide range of client and server platforms. Walter holds a degree in computer science from the University of Paderborn in Germany. His areas of expertise include consulting and project management in the area of MVS storage management and tape library implementation.

The authors of the first edition of this redbook were:

David Armes
IBM UK

Mike Broomhead
IBM UK

Yvonne Kratz
IBM Germany

Roger Stakkestad
IBM Norway

Leo SooHoo
IBM United States

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

Karen Dutch
IBM Storage Systems Division, San Jose

Dave Cannon
IBM Storage Systems Division, Tucson

Dave Crockett
IBM Storage Systems Division, Tucson

Rob Edwards
IBM Storage Systems Division, Endicott

Barry Fruchtman
IBM Storage Systems Division, Tucson

Mike Kaczmarski
IBM Storage Systems Division, Tucson

Maggie Cutler
International Technical Support Organization, San Jose Center

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in, “ITSO Redbook Evaluation” on page 369 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

  For Internet users`http://www.redbooks.ibm.com`
  For IBM Intranet users`http://w3.itso.ibm.com`
- Send us a note at the following address:

  `redbook@us.ibm.com`

Chapter 1. ADSM Version 3 Overview

ADSM Version 3 Overview

- Background
- ADSM Version 3
  - Client
  - Performance
  - Administration
  - Server
  - ADSMConnect Agents
- Summary

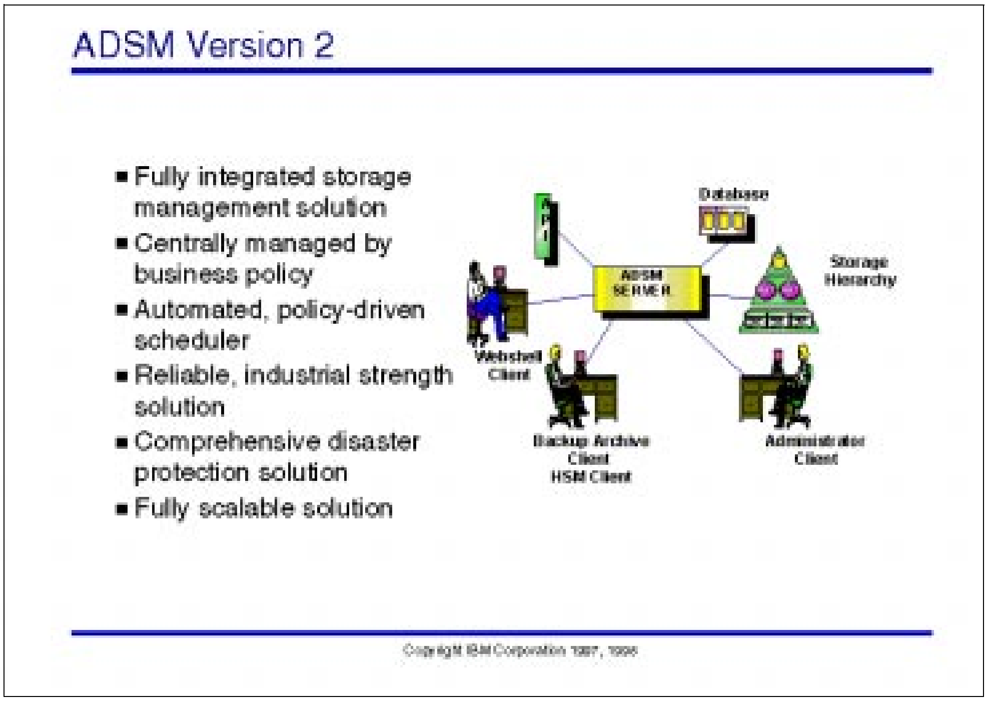
IBM
ADSTAR Distributed Storage Manager
Version 3

Copyright IBM Corporation 1997, 1998

This chapter provides an overview of ADSM Version 3. Topics covered are:

- Background, where it is explained why ADSM Version 3 has been developed
- ADSM Version 3 overview:
  - New client interfaces and functions
  - Server and client performance enhancements
  - Server administration enhancements
  - New server functions
  - ADSMConnect agents
- ADSM Version 3 Summary

1.1 Background



ADSM Version 2

- Fully integrated storage management solution
- Centrally managed by business policy
- Automated, policy-driven scheduler
- Reliable, industrial strength solution
- Comprehensive disaster protection solution
- Fully scalable solution

Copyright IBM Corporation 1997, 1998

1.1.1 ADSM Version 2

ADSM Version 2 was made available in 1995 and provided the following features:

1.1.1.1 Fully Integrated Storage Management Solution

ADSM Version 2 provided integration with many database applications such as DB2 and SAP and storage systems such as the IBM 3494 tape library.

1.1.1.2 Centrally Managed by Business Policy

ADSM Version 2 provided the capability to centrally manage storage requirements from distributed systems. Those requirements could be tailored to meet business requirements.

1.1.1.3 Automated, Policy-driven Scheduler

ADSM Version 2 provided an automated, policy-driven scheduler that provided customers with unattended data management services such as backup and archive.

1.1.1.4 Reliable, Industrial Strength Solution

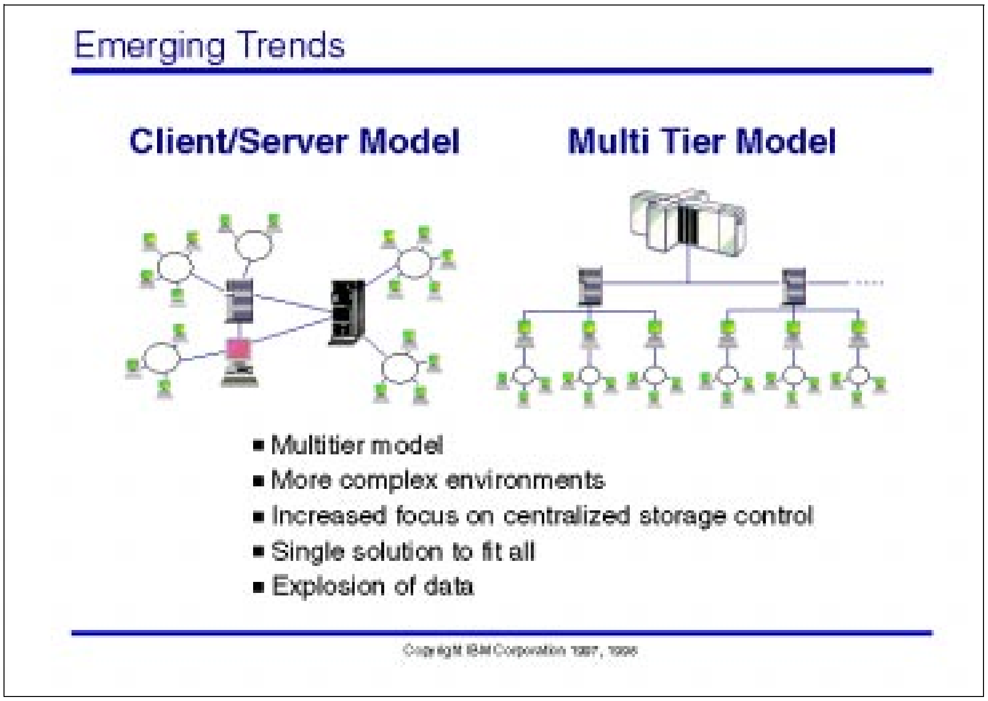
ADSM Version 2 provided a reliable, industrial strength storage management solution.

1.1.1.5 Comprehensive Disaster Protection Solution

ADSM Version 2 provided a comprehensive disaster protection solution with features such as copy storage pools and on-line database backup. An optional feature, Disaster Recovery Manager (DRM), provided automation for ADSM server disaster recovery.

1.1.1.6 Fully Scalable Solution

ADSM Version 2 was a fully scalable solution. An ADSM server could run on a PC, a UNIX server, an AS/400 server, or a mainframe.



Emerging Trends

Client/Server Model

Multi Tier Model

- Multitier model
- More complex environments
- Increased focus on centralized storage control
- Single solution to fit all
- Explosion of data

1.1.2 Emerging Trends

Since 1995 a number of trends have been observed in the IT industry. Many of these trends result from the popularity of the Internet and intranets and have significant data management requirements.

1.1.2.1 Multi Tier Model

Many companies are moving from a classic client/server model to a multitier model to expand the reach of IT services. This move is driven by the fact that company work forces are becoming more and more mobile, and business is spreading over a wider and wider area.

1.1.2.2 More Complex Environments

As client/server technologies mature, more complex environments are developing to provide data and services to new applications such as the Internet or to integrate with business technologies.

1.1.2.3 Increased Focus on Centralized Storage Control

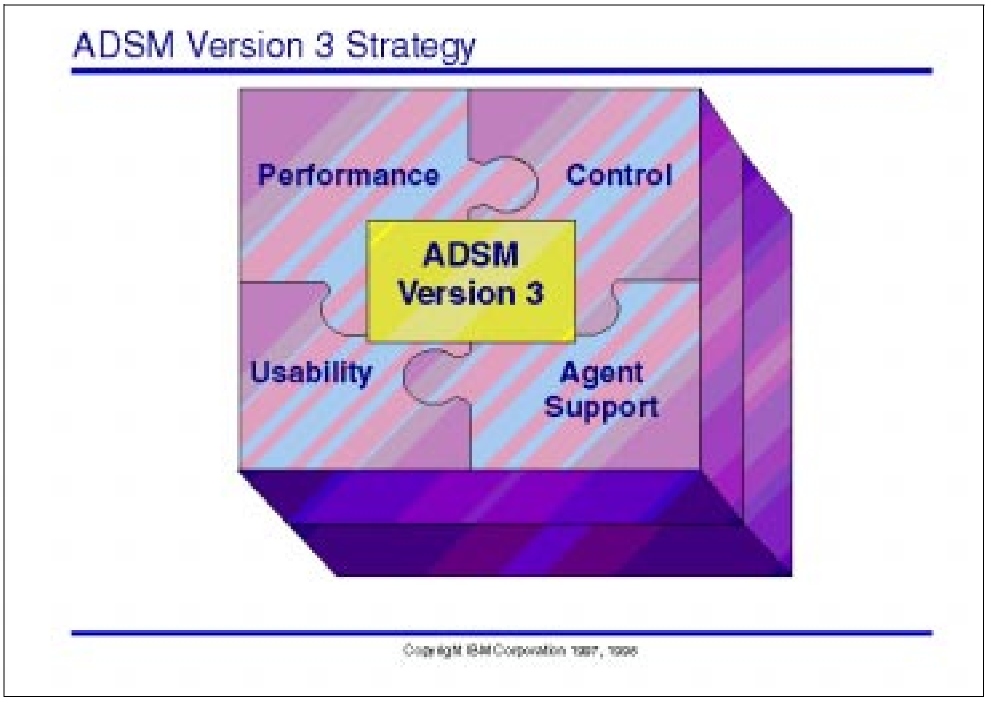
As distributed data is becoming more and more business critical and network costs fall, the focus on centralized storage control increases.

1.1.2.4 Single Solution to Fit All

Although enterprises may implement solutions based on different hardware, they want a consistent, single solution for all systems management requirements, including storage management.

1.1.2.5 Explosion of Data

The growth in distributed data has been enormous. This growth is partly due to lower network and hardware costs and new applications such as databases and Internet solutions.



ADSM Version 3 Strategy

Performance

Control

ADSM Version 3

Usability

Agent Support

Copyright IBM Corporation 1997, 1998

1.1.3 ADSM Version 3 Strategy

A strategy for ADSM Version 3 has been created to build on the strengths of ADSM Version 2 and address new challenges from emerging trends.

1.1.3.1 Performance

ADSM Version 3 addresses performance because of the explosion of data. New clients, protocols, and functions have been introduced.

1.1.3.2 Control

Better control to meet the needs of complex environments, including multitier client/server, is enabled with the introduction of ADSM Version 3.

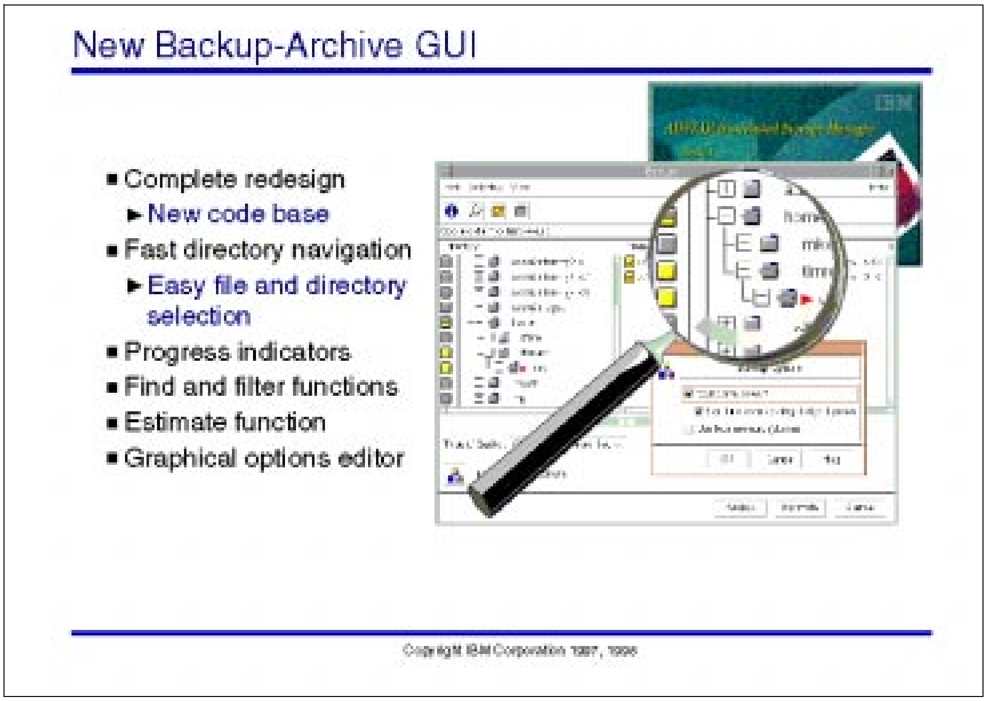
1.1.3.3 Usability

The usability of ADSM has been improved, both for administrators and end users.

1.1.3.4 Agent Support

ADSMConnect agents integrate ADSM with distributed data applications to provide ADSM services.

1.2 ADSM Version 3 Overview



New Backup-Archive GUI

- Complete redesign
  - ► New code base
- Fast directory navigation
  - ► Easy file and directory selection
- Progress indicators
- Find and filter functions
- Estimate function
- Graphical options editor

Copyright IBM Corporation 1997, 1998

1.2.1 New Backup-Archive GUI

On the basis of extensive usability studies, the ADSM backup-archive client GUI has been rewritten for ADSM Version 3. The new GUI offers an easy-to-use and intuitive way of performing functions such as backup and restore.

1.2.1.1 Complete Redesign

The backup-archive client GUI uses a completely new code base. The UNIX backup-archive client interface has been coded with Motif libraries, and the OS/2 backup-archive client is coded with the 32-bit OS/2 APIs.

1.2.1.2 Fast Directory Navigation

The backup-archive client is designed to manage client filespaces containing a large number of files. Detailed file and directory information is not requested or displayed until needed; for example, in a backup operation, detailed directory information is not read from the client disk until the directory is expanded.

Selecting directories of files for backup is made easy by providing a selection symbol in front of all directories and files displayed.

1.2.1.3 Progress Indicators

Progress indicators are added to the GUI to show a user that ADSM activity is occurring, for example, files are being scanned or backed up.

1.2.1.4 Find and Filter Functions

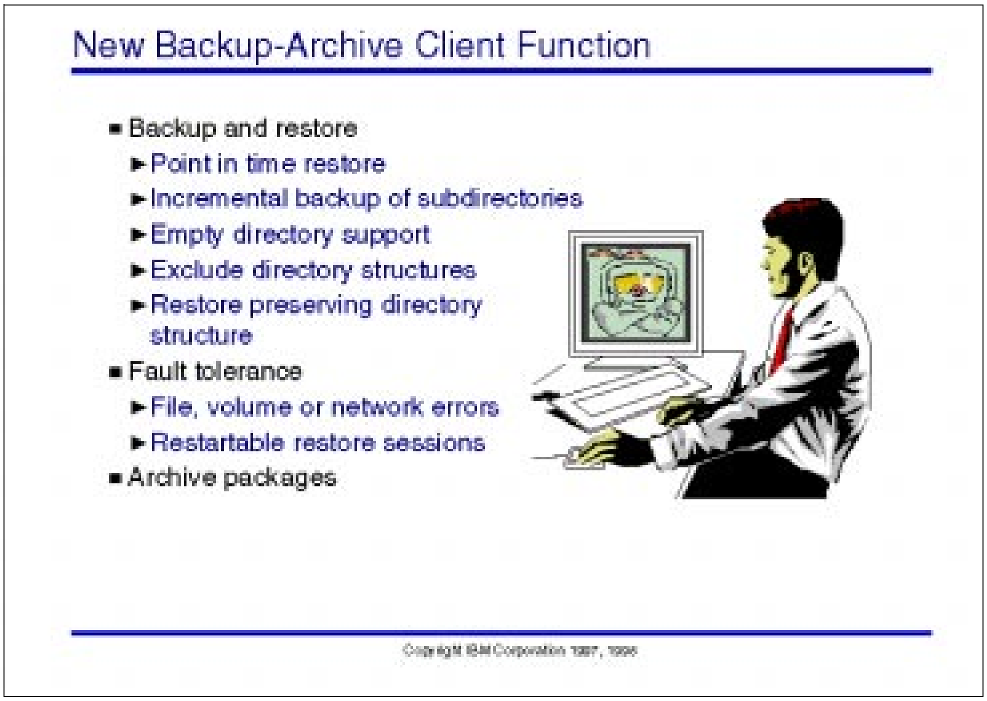
The Find function filters or searches for files stored locally on a client’s disk or managed by ADSM. Invoking the Find function brings up a window to filter files or search for files matching certain criteria.

1.2.1.5 Estimate Function

The ADSM Version 3 backup-archive client GUI allows an Estimate function to be performed before the start of an ADSM operation. The ADSM backup-archive client estimates the elapsed time for the operation on the basis of historical information. A user can choose to cancel the operation before it starts if the amount of data selected or the estimated elapsed time for the operation is excessive.

1.2.1.6 Graphical Options Editor

A new graphical options editor is provided with the ADSM Version 3 backup-archive GUI. The graphical options editor makes the process of updating the ADSM client options more user friendly and less error prone than before. The graphical options editor updates the client configuration files, dsm.opt and dsm.sys (UNIX clients only), if any options are changed. The client configuration files can still be updated by using a text editor on the clients.



New Backup-Archive Client Function

- Backup and restore
  - Point in time restore
  - Incremental backup of subdirectories
  - Empty directory support
  - Exclude directory structures
  - Restore preserving directory structure
- Fault tolerance
  - File, volume or network errors
  - Restartable restore sessions
- Archive packages

Copyright IBM Corporation 1997, 1998

1.2.2 New Backup-Archive Client Function

New function has been added to the ADSM backup-archive client.

1.2.2.1 Backup and Restore

The backup and restore function has been enhanced with several new features.

A point-in-time restore restores a filespace (or directory) to the state it had prior to a given date and time. Support for point-in-time restore is essential for recovering a filespace or directory to a time when it was known to be in a good or consistent state. In ADSM Version 3 point-in-time restores are supported at the filespace and directory level. Both the backup-archive GUI and command line clients support point-in-time restore.

In ADSM Version 2 an incremental backup is performed at the filespace level (file system, partition, or volume). The ADSM Version 3 backup-archive client supports incremental backup of directories. Incremental backup on the directory level can greatly reduce the amount of data that is backed up and the time it takes to perform the backup. The new support for incremental backup of directory structures is invoked automatically if a directory is selected from the backup-archive GUI or command line client and incremental backup is selected.

The ADSM Version 3 backup-archive client selective backup has been enhanced to support backup and restore of empty directory structures. All directory attributes such as ACL, trustee rights, and file permissions are supported.

When a directory structure is excluded from backup by an EXCLUDE include-exclude statement, the directory objects of the excluded directory structure are backed up. Only the files in an excluded directory structure are excluded from backup. A new include-exclude statement, EXCLUDE.DIR, is introduced in ADSM Version 3. The EXCLUDE.DIR statement excludes a directory structure and its attributes from being backed up.

Preserving directory structures when restoring files to a new location is difficult with ADSM Version 2. This problem has been addressed in the ADSM Version 3 backup-archive client. The ADSM Version 3 client can restore a directory structure using three different parameters:

- Complete path
- Restore partial path
- Do not preserve directory structure

1.2.2.2 Fault Tolerance

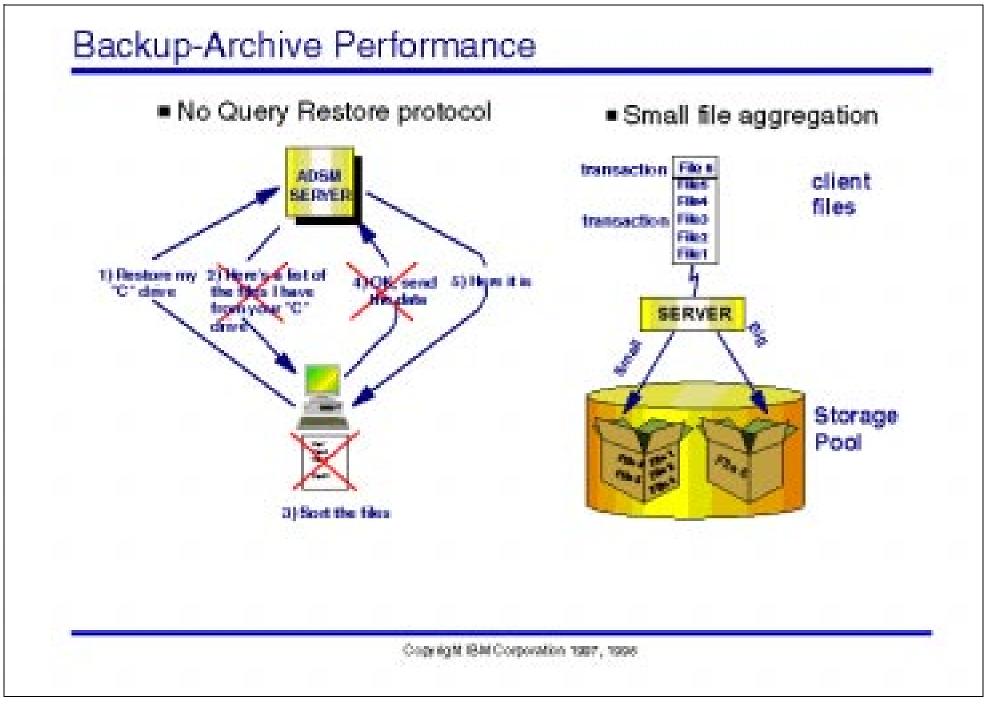
ADSM Version 3 incorporates fault-tolerant characteristics into clients. The goal is to allow for survival of client operations after media or network failures.

The backup-archive client is enhanced to handle file errors on the client and storage volume errors on the server by restarting the operation, beginning with the failing file. If communication errors occur during a backup or restore operation, the client attempts to reopen the session.

Fault tolerance is further enhanced with the concept of restartable restores. Restartable restores allow failed restore operations to be restarted from the point of failure.

1.2.2.3 Archive Packages

The association between archived files and their descriptions is much stronger in ADSM Version 3 than in previous versions. All files archived require a description in ADSM Version 3. An archive package is a set of files and directories archived with a common, unique archive description.The ADSM Version 3 GUI retrieve function displays archived files hierarchically in a collapsible directory tree. The files are grouped by their archive descriptions. Expanding the collapsible description tree displays the individual directories and files that the archive package contains.



Backup-Archive Performance

- No Query Restore protocol
- Small file aggregation

1.2.3 Backup-Archive Performance

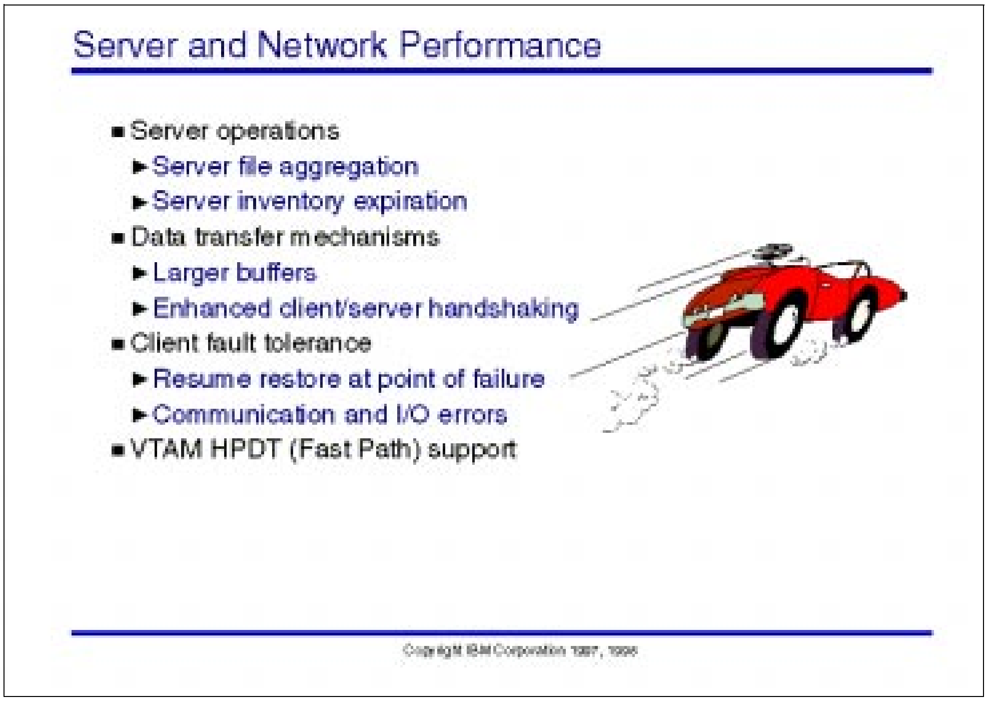
ADSM Version 3 introduces new client/server protocols and functions.

1.2.3.1 No Query Restore Protocol

An additional restore protocol called *No Query Restore* is introduced to assist in removing client memory constraints during the restore process. The previous restore protocol relied on the client to build a list of files and pull them from the server. The No Query Restore protocol eliminates client building and sorting of files by having the server send files following a restore request from a client, reducing client memory requirements. This allows restores of large amounts of files that had previously failed because the creation of the list of files on the client required more memory than was available. Clients such as NetWare which have memory resource constraints, many directories, or many files to restore, will receive the most benefit. Removing the file list sorting from the client could also improve overall performance for the restore operation. The No Query Restore protocol enables restores to be restartable if the restore process is interrupted before completion because the ADSM server maintains the list of files being restored.

1.2.3.2 Small File Aggregation

Small file aggregation is a new server function that groups small client files in larger physical files. This function will significantly improve client/server operations with small files. Aggregates are based on transactions to the ADSM server, a unit of committed work. As always, file granularity is maintained within ADSM. Backup-archive sessions have shown performance improvements as a result of aggregating small files. The greater the number of small files that are aggregated, the fewer the number of database transactions when compared with previous versions of ADSM without aggregation.



Server and Network Performance

- Server operations
  - Server file aggregation
  - Server inventory expiration
- Data transfer mechanisms
  - Larger buffers
  - Enhanced client/server handshaking
- Client fault tolerance
  - Resume restore at point of failure
  - Communication and I/O errors
- VTAM HPDT (Fast Path) support

Copyright IBM Corporation 1997, 1998

1.2.4 Server and Network Performance

ADSM Version 3 implements a number of enhancements and new features to further improve performance on the server and in the communications layer.

1.2.4.1 Server Operations

For server performance, processes that are a function of file size are affected most by server file aggregation. Server processes that involve copy or move operations such as storage pool migration, storage pool backup and restore, and the move data operation are the primary beneficiaries of aggregation. The grouping of small client files into large physical files enables server processes to execute faster than previous versions of ADSM because of fewer database transactions. The database entries for a logical file within an aggregate are less than the entries for a single physical file. Compared with previous versions, which required database entries for every file, this reduction in entries results in a smaller ADSM database.

The inventory expiration process has a new optimized algorithm. In addition to other ADSM database code improvements, this new algorithm is designed to provide improvements in the expiration process.

1.2.4.2 Data Transfer Mechanisms

The use of larger buffers on the server is supported in all ADSM Version 3 servers. Thus larger communication buffers are supported when the server communicates with a client, and the performance in client/server communications is improved. Larger device I/O buffers are also supported, so disk I/O resources are used more efficiently. Reads and writes are quicker and server resources are better utilized, including lower CPU utilization. By reducing CPU utilization, more clients can concurrently be serviced, improving overall system performance. The benefits from the use of server aggregation and larger buffers are complementary. Aggregation groups smaller, logical client files into fewer but larger physical files at the server level. Larger files are better able to take advantage of the larger buffers.

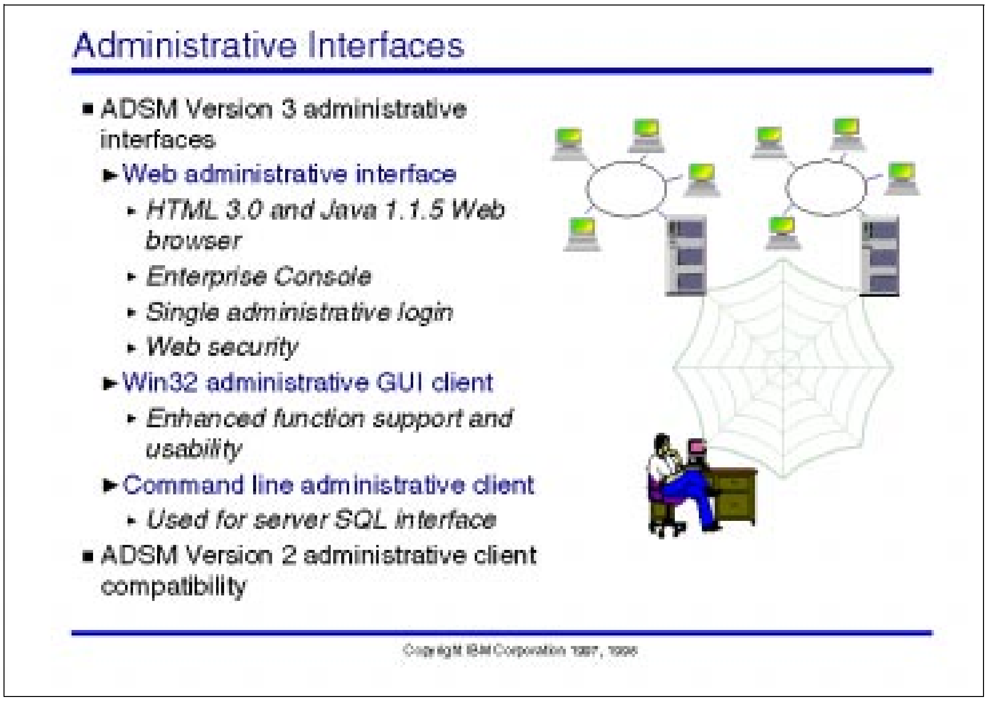
Client/server communications within the ADSM application have been modified to use enhanced client/server confirms, or handshaking, during communications, thereby reducing the overhead for transactions.

1.2.4.3 Client Fault Tolerance

Although these items are not specifically designed to improve raw performance throughput, they do assist in improving the manageability of a system and in the overall completion of client tasks, such the backup of a client’s filespace. These features help to reduce overall task completion time in situations where error conditions or interruptions require detection, correction, intervention, and repetition of data movement. For example, if client filespace restore is interrupted in the middle of a restore process, the restore could be restarted without having to restart from the beginning. Fault tolerance is extended to handle situations requiring restartable restores, communication interruption handling, and addressing files in error such as file or volume errors.

1.2.4.4 VTAM HPDT Support

Support for the new VTAM 4.4 High Performance Data Transfer (HPDT or Fast Path) is added for MVS servers using the APPC communications method. This support helps efficiently manage copies of data between VTAM and ADSM.



Administrative Interfaces

- ADSM Version 3 administrative interfaces
  - Web administrative interface
    - *HTML 3.0 and Java 1.1.5 Web browser*
    - *Enterprise Console*
    - *Single administrative login*
    - *Web security*
  - Win32 administrative GUI client
    - *Enhanced function support and usability*
  - Command line administrative client
    - *Used for server SQL interface*
- ADSM Version 2 administrative client compatibility

Copyright IBM Corporation 1997, 1998

1.2.5 Administrative Interfaces

The ADSM administrative clients are the administrator’s interface to the server and as such are key to the usability of ADSM. ADSM Version 3 expands on the function available in the ADSM Version 2 administrative GUIs and adds support for new server functions.

1.2.5.1 ADSM Version 3 Administrative Interfaces

ADSM Version 3 does not have administrative GUI clients for all client platforms that were supported in ADSM Version 2. With the exception of the Win32 client, all of the administrative GUI clients have been replaced by the new Web administrative interface. This leaves three choices of interface for the ADSM administrator:

- Web administrative interface - The Web administrative interface is the standard for ADSM Version 3. Thus the ADSM administrator can become familiar with one interface and use that interface to control all ADSM server platforms within the enterprise. The Web-based nature of the interface allows control of the ADSM server from any workstation capable of running an HTML 3.0 and Java 1.1.5 or higher compliant Web browser. The Web administrative interface is a server function and is available only with ADSM Version 3 servers.

  The Web administrative interface provides an Enterprise Console for administering multiple ADSM servers and clients and enables the ADSM administrator to manage all ADSM functions, through a combination of the GUI, a command line interpreter, and an event viewer. The controls available have been extended from those available in the ADSM Version 2 administrative GUI and include all ADSM Version 2 functions and the new functions introduced in ADSM Version 3.

The Web administrative interface provides a single administrative login capability. Thus administrators can log in once to ADSM and then navigate from one ADSM server to any other suitably configured ADSM servers and clients without having to reenter their administrator userids and password.

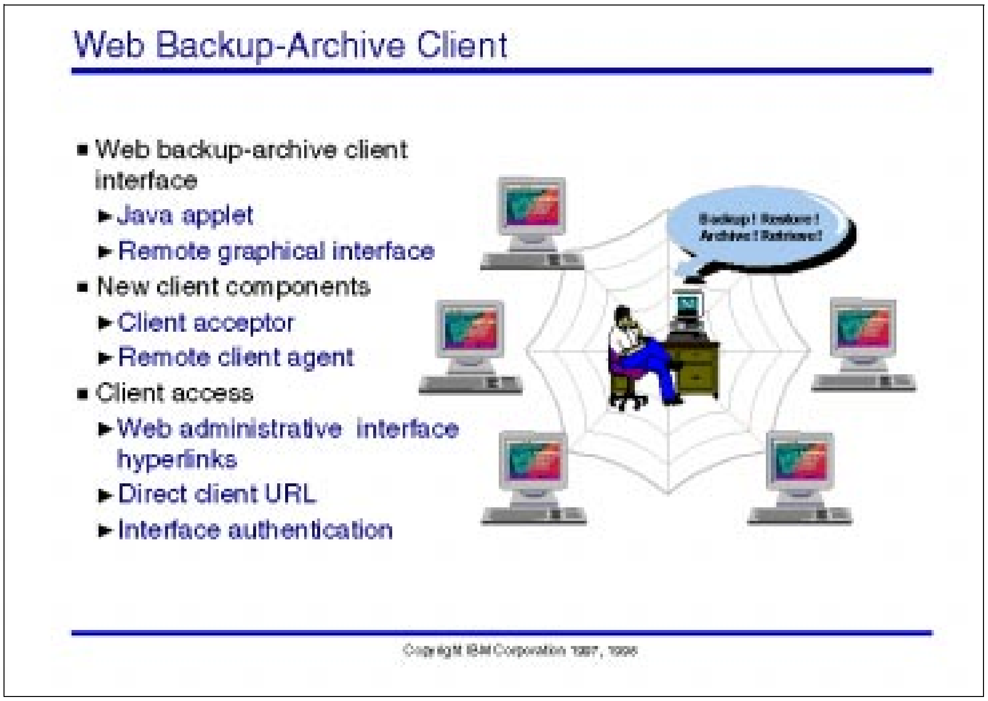
A number of additional functions have been added to ADSM Version 3 to enhance the security of the Web administrative interface. These functions include support for Secure Sockets Layer (SSL) communications, enhanced logging of Web-based sessions, password restrictions, and client or administrator lockout capabilities.

- Win32 administrative GUI client - The Win32 administrative GUI has had a number of modifications in the way the client interfaces with the operating system and the way that the user interfaces with the client. The modifications have led to improvements in performance and usability of the client interface, making it more user friendly and improving performance. The Win32 administrative GUI now uses many of the features of modern GUI design, including context-sensitive menus, more use of the right mouse button, and an OK button to accept changes and close a subwindow. The function support has also been broadened, and many of the ADSM Version 3 enhancements are now configured through this interface.
- Command line administrative client - The command line administrative client is available for all of the same client platforms as ADSM Version 2. The command line interface is the primary interface for using the new server SQL interface.

1.2.5.2 ADSM Version 2 Administrative Client Compatibility

Existing ADSM Version 2 administrative command line and GUI clients can be used to administer a ADSM Version 3 server. The command line clients can be used for all ADSM Version 3 functions, but the GUI client only supports existing ADSM Version 2 functions.

The ADSM Version 3 administrative clients, with the exception of the Web administrative, which is a server function, can be used to administer existing ADSM Version 2 servers. Icons in the ADSM Version 3 GUI administrative client that relate to new ADSM Version 3 function will not be available when used with a ADSM Version 2 server.



Web Backup-Archive Client

- Web backup-archive client interface
  - ► Java applet
  - ► Remote graphical interface
- New client components
  - ► Client acceptor
  - ► Remote client agent
- Client access
  - ► Web administrative interface hyperlinks
  - ► Direct client URL
  - ► Interface authentication

Backup! Restore! Archive! Retrieve!

1.2.6 Web Backup-Archive Client

ADSM Version 3.1.2 provides a new Web backup-archive client interface to perform remote client operations.

1.2.6.1 Web Backup-Archive Client Interface

À centralized administrative approach for ADSM, such as a helpdesk, requires that an administrator perform remote client operations. The new Web backup-archive client interface is a Java applet that provides an easy-to-use, intuitive way of performing client actions such as backups and restores.

This new interface can be used with any Web browser with support for HTML 3.0 and Java 1.1.5. The Web client is invoked by pointing the Web browser at the URL for the client workstation. The Java applet is served from the client workstation to the user's Web browser and provides a similar GUI to that of the existing backup-archive GUI client.

1.2.6.2 New Client Components

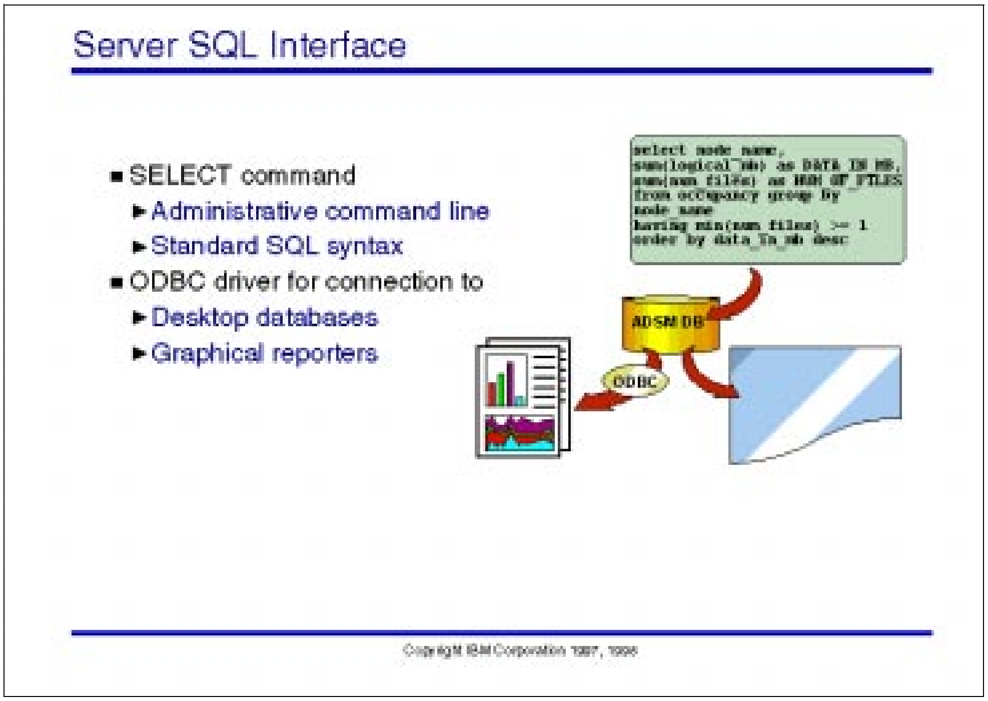
The Web client is installed with the backup-archive client package and consists of two new processes on the client workstation: the client acceptor and remote client agent.

The client acceptor is an HTTP daemon that serves the Web client Java applet to the Web browser. The remote client agent performs the client functions initiated with the Web client interface.

1.2.6.3 Client Access

There are two methods of accessing the Web client from a Web browser. One method involves creating a hyperlink from the Web administrative Enterprise Console by defining a URL for the Web client. Defining a URL creates a client hyperlink icon. Clicking on this hyperlink icon connects to the Web client. The other method is to connect directly from a Web browser by entering the URL and port number of the client workstation in the location field of the browser.

An administrator userid and password are required for authentication whenever client backup, restore, archive, or retrieve functions are initiated. If the Web client connection is made from the Web administrative Enterprise Console, the single administrative login feature is exploited and the authentication is transparent, assuming that the administrative userid being used has sufficient authority. If the Web client connection is established through a direct client URL, the user is prompted for an administrator userid and password.



Server SQL Interface

- SELECT command
  - Administrative command line
  - Standard SQL syntax
- ODBC driver for connection to
  - Desktop databases
  - Graphical reporters

```
select node_name,
sum(logical_mb) as DATA_IN_MB,
sum(num_files) as NUM_OF_FILES
from occupancy group by
node_name
having min(num_files) >= 1
order by data_in_mb desc
```

1.2.7 Server SQL Interface

With ADSM Version 2, information stored in the ADSM database was only available through a set of query commands. These commands produced preformatted output with little opportunity for customized reports. ADSM Version 3 addresses this requirement by providing an SQL interface. The interface is read only and includes a SELECT command and an ODBC driver on Windows NT.

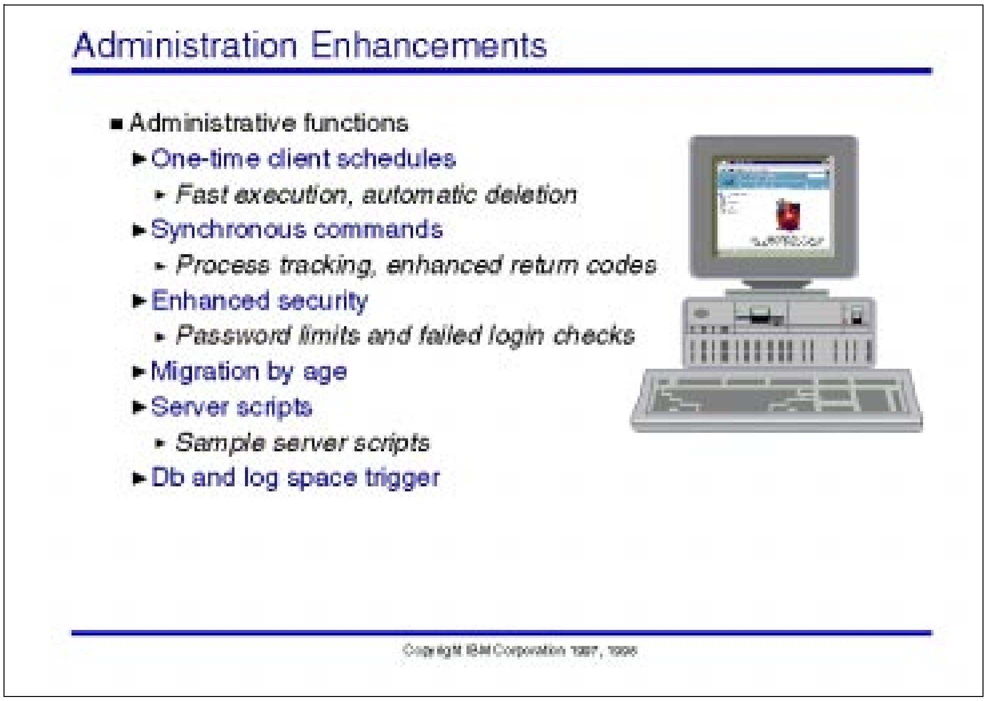
1.2.7.1 SELECT Command

The SQL interface represents ADSM information in the form of relational tables containing rows and columns that can be accessed by the SELECT command. The SELECT command uses standard SQL syntax compliant with the SQL92/93 standard and can be used only on the administrative command line client.

Since SQL processing uses ADSM database resources, long-running or very complicated select statements can slow down server performance significantly. Therefore resource-intensive queries display a confirmation message, offering the possibility to abort the query before executing it.

1.2.7.2 ODBC Driver

ODBC is a standard interface between SQL database engines and front-end applications. It allows products such as Lotus Approach or Microsoft Access to be used to graphically construct SQL queries, which are then dispatched to the database (in this case the ADSM database). The select statement results are returned in tabular form and can be processed to be displayed as graphs or tables. The ADSM ODBC driver ships as part of the client package and is only available for the Win32 client. The driver is compliant with the ODBC 2.5 API.



Administration Enhancements

- Administrative functions
  - One-time client schedules
    - *Fast execution, automatic deletion*
  - Synchronous commands
    - *Process tracking, enhanced return codes*
  - Enhanced security
    - *Password limits and failed login checks*
  - Migration by age
  - Server scripts
    - *Sample server scripts*
  - Db and log space trigger

Copyright IBM Corporation 1997, 1998

1.2.8 Administration Enhancements

In addition to administration enhancements in ADSM Version 3 for new functions, more general administration support is delivered. The aim of the enhancements is to make the life of the ADSM administrator easier and the tasks performed more intuitive.

1.2.8.1 Administrative Functions

In ADSM Version 3, a new command has been introduced to enable administrators to execute one-time client-scheduled commands. DEFINE CLIENTACTION allows an ADSM administrator to have a command executed on an ADSM client, without defining a schedule and associating the schedule with the client. The new command allows an administrator to have a client execute a command one time only, and then have the schedule and associations deleted from the database automatically.

A new parameter has been added to many commands that in ADSM Version 2 would have started background processes, forcing them to run in the foreground and concurrently report any messages including a message and return code on completion. The WAIT=YES option runs the process in the foreground, reports messages concurrently, and returns to the issuing session only on completion. Thus an ADSM administrator can issue a command and have real-time reporting on its progress along with a meaningful message on completion.

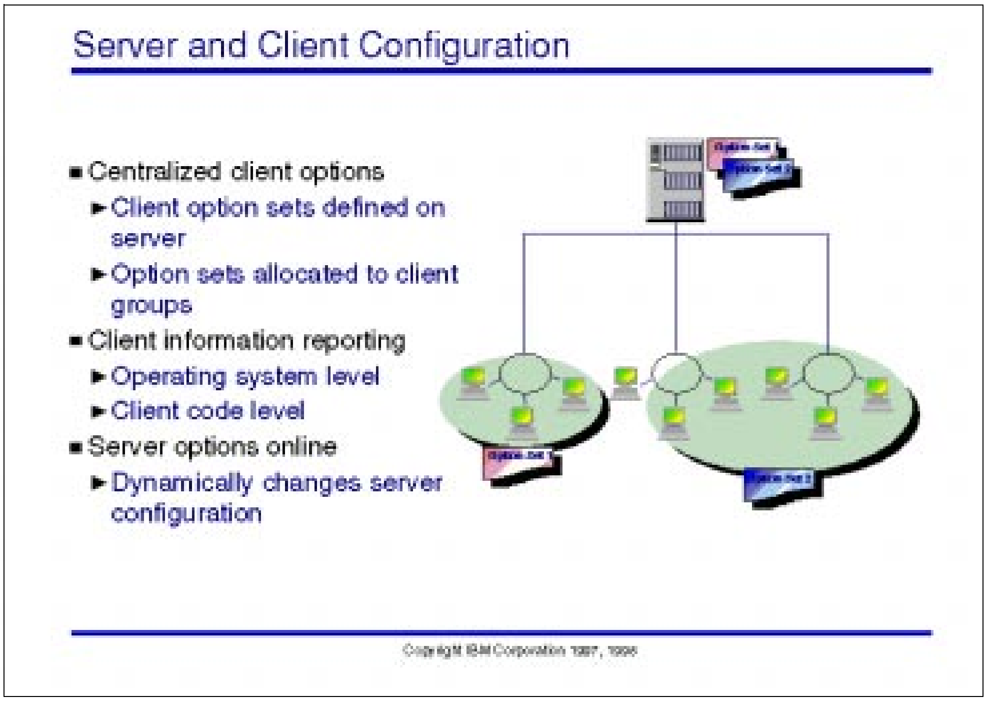
Server security has also been enhanced in ADSM Version 3, and a record is kept of the number of consecutive unsuccessful Log in attempts by any node or administrator. A counter increments each time a client or administrator fails sign on authentication and is reset to 0 when sign on completes successfully. The ADSM administrator can also specify the number of consecutive invalid attempts nodes or administrators can have before they are locked out of the ADSM server.

Other security enhancements have also been implemented. It is now possible to set the minimum length of ADSM passwords and to force a node or administrator to reset his or her password on next server access.

With ADSM Version 3 new *migdelay* and *migcontinue* parameters have been added to the storage pool definition. These parameters enable an administrator to specify a minimum number of days data should remain in a storage pool before it is migrated to the next pool in the hierarchy. Thus there is another level of control, in addition to migration by threshold, for the movement of data within the storage pool hierarchy.

Server automation and management operations have been enhanced by the use of server scripts for execution of administrative commands. Server scripts are defined and stored in the ADSM database. They contain administrative commands that are executed sequentially. The execution of those commands can be controlled with variables, logic flow statements, and return codes. The defined script can be scheduled by using the new RUN administrative command. A set of sample scripts is provided with the ADSM server.

A space trigger can be defined for the server database and recovery log. When this trigger is reached, ADSM automatically extends the database or recovery log space, using predefined values. This new function helps prevent out-of-space conditions for the database and the recovery log.



Server and Client Configuration

- Centralized client options
  - Client option sets defined on server
  - Option sets allocated to client groups
- Client information reporting
  - Operating system level
  - Client code level
- Server options online
  - Dynamically changes server configuration

Copyright IBM Corporation 1997, 1998

1.2.9 Server and Client Configuration

As the number of clients and options in an ADSM environment grows, it becomes increasingly important that they can be configured centrally. ADSM Version 3 introduces new functions to assist in the central management of clients.

1.2.9.1 Centralized Client Options

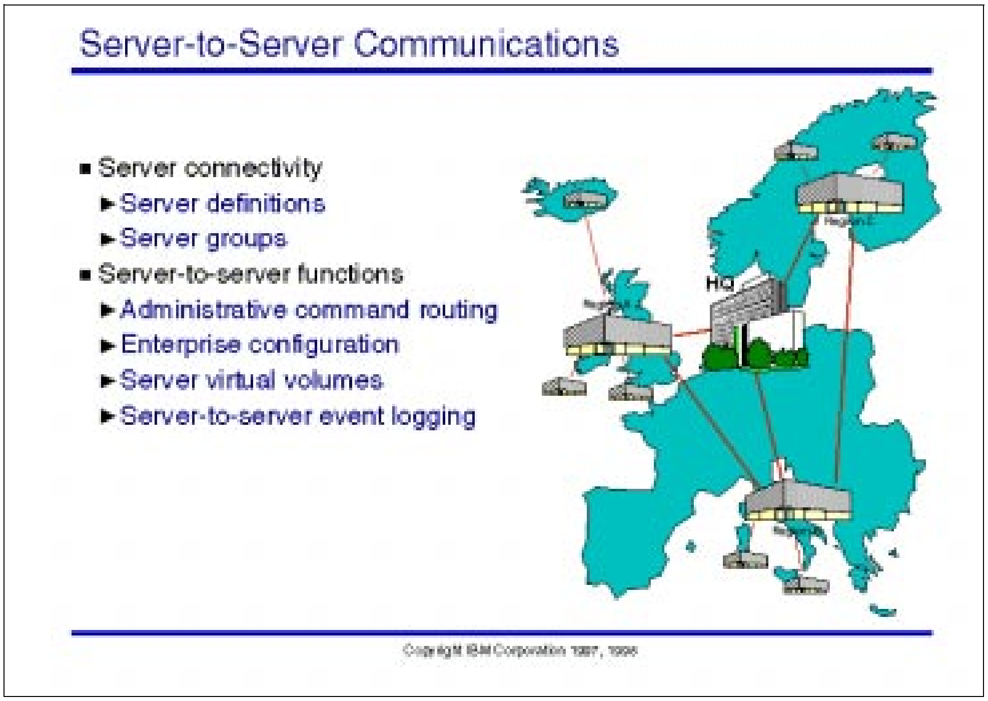
ADSM Version 3 introduces centralized client options. Client options can be defined in *option sets* on the server, and client nodes are associated with the option set. The association can be made at initial registration of the client node or at any time during ADSM operation. Thus nodes with similar option requirements can be grouped, and large groups of nodes can have their options updated simultaneously, without the need to access the individual machines or edit ASCII text files. Because administrators can lock certain client options, system wide option policies can be enforced.

1.2.9.2 Client Information Reporting

In ADSM Version 2, the server stored some information about the client node’s operating system. This information was passed to the server at initial sign on and never updated. In ADSM Version 3 this information has been expanded to include operating system level and the level of ADSM client code. Thus administrators can query the nodes to gain information about the various ADSM client and operating system levels in their environment.

1.2.9.3 Server Options On-line

In ADSM Version 3 new function has been added to enable an ADSM administrator to change certain server options on-line, removing the need to halt and restart the ADSM server when options have to be changed.



Server-to-Server Communications

- Server connectivity
  - Server definitions
  - Server groups
- Server-to-server functions
  - Administrative command routing
  - Enterprise configuration
  - Server virtual volumes
  - Server-to-server event logging

Copyright IBM Corporation 1997, 1998

1.2.10 Server-to-Server Communications

ADSM Version 3 introduced server-to-server communications for interconnecting multiple ADSM servers. ADSM Version 3.1.2 expands this support and provides new functions that exploit this enterprisewide connectivity.

1.2.10.1 Server Connectivity

Server-to-server communications provides the mechanism for creating network connections between ADSM servers. Server-to-server communications uses TCP/IP as the communication protocol between servers. Servers are defined with a server name, a server password, and the server TCP/IP address. The server name and password are used to authenticate sessions initiated by a remote server.

1.2.10.2 Server-to-Server Functions

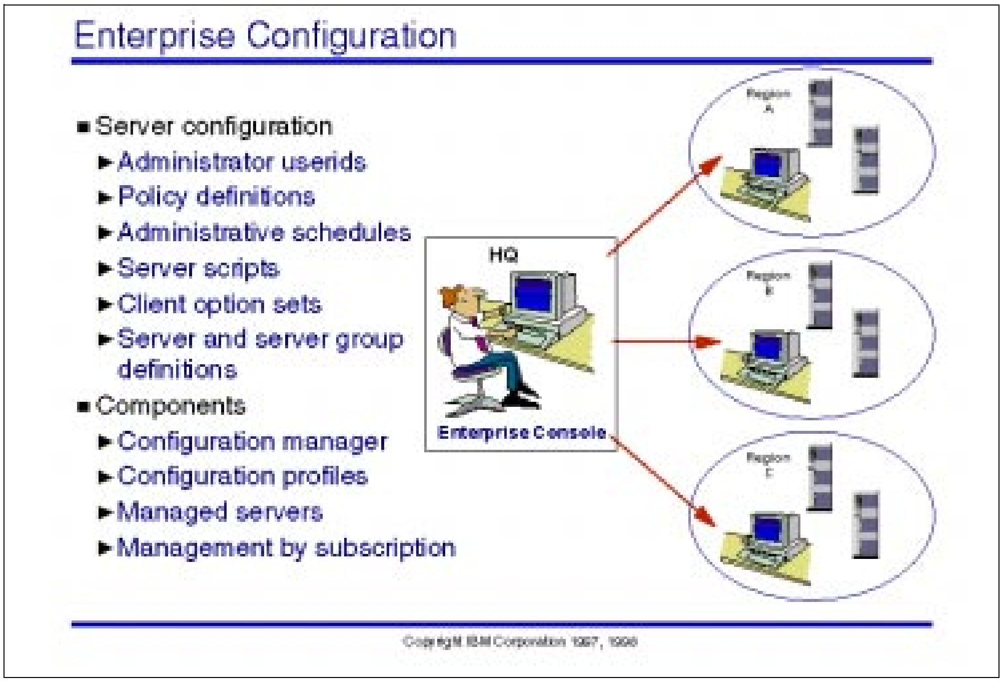
Server-to-server communications are used for the administrative command routing, enterprise configuration, server virtual volumes, and server-to-server event logging functions.

Administrative command routing allows administrators to issue administrative commands from one server and route them to other target servers. The commands are executed on the target servers, and the command output is returned and formatted on the server where the command was issued.

Enterprise configuration allows server configurations to be defined centrally by an administrator and then propagated to other servers. This simplifies the configuration and management of enterprises with multiple ADSM servers.

Virtual volumes can be defined on ADSM servers. These volumes can be used for primary or copy storage pools and for server database backups. The data contained within the virtual volume is physically stored on another network-connected ADSM server.

In an enterprise environment with multiple ADSM servers, client and server events can be logged to a central management server through server-to-server communications, thereby enabling centralized event management and automation.



Enterprise Configuration

- Server configuration
  - Administrator userids
  - Policy definitions
  - Administrative schedules
  - Server scripts
  - Client option sets
  - Server and server group definitions
- Components
  - Configuration manager
  - Configuration profiles
  - Managed servers
  - Management by subscription

Copyright IBM Corporation 1997, 1998

1.2.11 Enterprise Configuration

Enterprise configuration allows server configurations to be defined centrally by an administrator and then propagated to other servers. This simplifies the configuration and management of multiple ADSM servers in an enterprise.

1.2.11.1 Server Configuration

With enterprise configuration the following server configuration objects can be defined centrally on an ADSM server:

- Administrator userids
- Policy definitions
- Administrative schedules
- Server scripts
- Client option sets
- Server and server group definitions

These centrally defined configuration objects are then automatically copied to other ADSM servers. Whenever a new configuration object is added or an existing object is modified or deleted, the copies on the other ADSM servers are synchronized automatically.

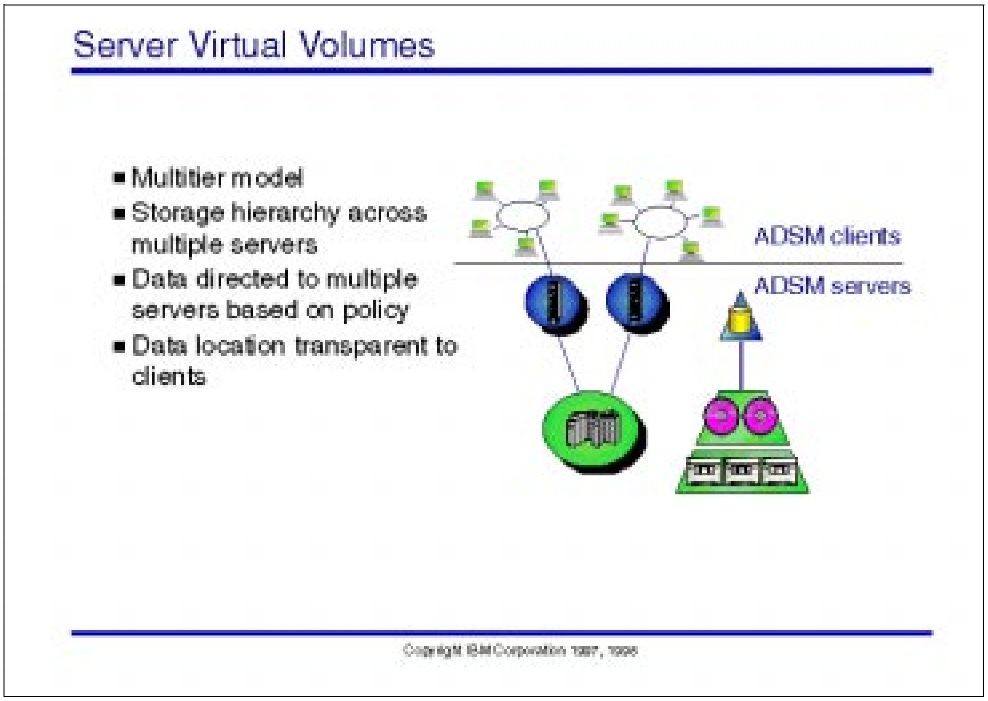
1.2.11.2 Components

Enterprise configuration introduces new terminology and system components. A configuration manager is an ADSM server where a central configuration is defined that will be propagated to other servers.

A configuration profile is a set of definitions on the configuration manager that define the objects on the configuration manager that will be propagated to the other servers. The profile contains references to the objects defined on the configuration manager, not the actual objects.

The servers that receive copies of the configuration objects are called *managed servers*. They are managed because their configuration, or a subset of their configuration, is defined centrally on a configuration manager. These managed objects on the managed server cannot be modified or deleted locally. Changes can be made only on the configuration manager. The changes are automatically propagated.

A server becomes managed by subscribing to a configuration profile defined on a configuration manager server. When a server subscribes to a configuration profile, the objects associated with the profile are copied from the database on the configuration manager server to the database of the managed server. The managed server periodically contacts, or polls, the configuration manager server to determine whether the profiles, and associated objects, to which the managed server is subscribed have been modified. If they have, the updates are automatically copied to the managed server.



Server Virtual Volumes

- Multitier model
- Storage hierarchy across multiple servers
- Data directed to multiple servers based on policy
- Data location transparent to clients

Copyright IBM Corporation 1997, 1998

1.2.12 Server-to-Server Virtual Volumes

ADSM Version 3 introduces server-to-server virtual volumes. Thus one ADSM server can store data on another ADSM server. This capability increases the flexibility in designing ADSM configurations.

1.2.12.1 Multitier Model

There are several variations of a multitier model, including a branch office model, collection of workgroups, or collection of distributed LANs. ADSM server-to-server virtual volumes can meet the data management requirements of these models by making best use of network and hardware resources and providing site disaster protection.

1.2.12.2 Storage Hierarchy across Multiple Servers

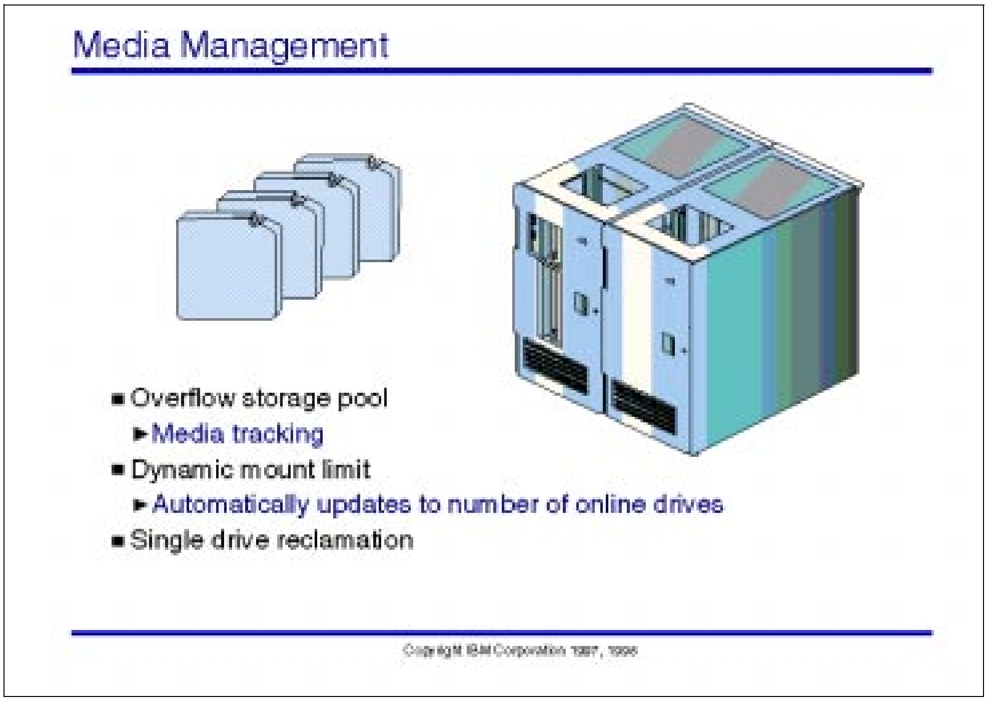
An existing ADSM storage hierarchy can easily be extended to a server hierarchy. This new capability allows departmental ADSM servers to share central storage resources such as tape libraries.

1.2.12.3 Data Directed to Multiple Servers Based on Policy

Different data of ADSM clients can be bound to different management classes on the local ADSM server. These management classes can direct the data to storage pools on that server or on a different server. When client workstations back up files that are bound to such management classes, a single client session backs up data to multiple ADSM servers.

1.2.12.4 Data Location Transparent to Clients

Just as ADSM clients do not need to know which storage pool volumes hold their data, they also do not need to know which ADSM server holds their data. However, a client always uses the same ADSM server as a single point of contact. Only that server has the necessary database entries for that client’s data. The same ADSM server to which the client backed up must be available for the client to perform a restore.



Media Management

- Overflow storage pool
  - Media tracking
- Dynamic mount limit
  - Automatically updates to number of online drives
- Single drive reclamation

Copyright IBM Corporation 1997, 1998

1.2.13 Media Management

Because ADSM is integrated with a large number of storage devices, control of media volumes is a crucial task of administrators. To make this task less labor intensive, ADSM Version 3 provides a number of new functions and functional enhancements.

1.2.13.1 Overflow Storage Pool

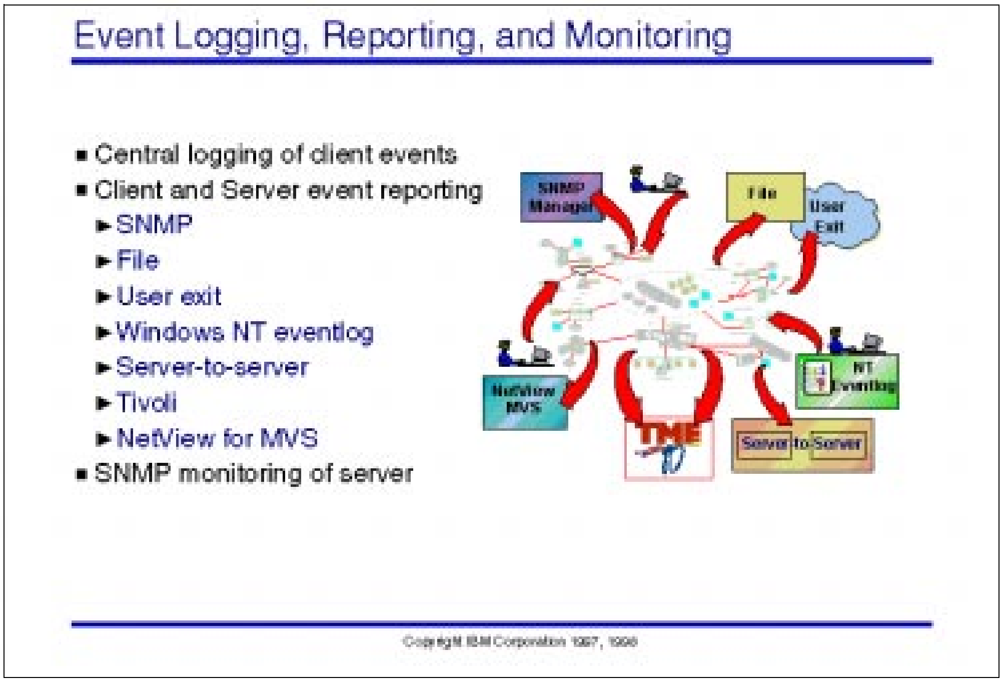
Overflow storage pools are introduced to aid administrators in the management of sequential media storage pools that contain more volumes than can be held in a library. Primarily for long-term archive pools, this function integrates tracking mechanisms into the server to manage volume state and location. Overflow storage pool volumes do not replace DRM volumes because they are designed to remain onsite and available for ADSM use.

1.2.13.2 Dynamic Mount Limit

Dynamic mount limit management in ADSM Version 3 allows the server to monitor the number of drives that are available for ADSM use and adjust the device class settings accordingly. Thus it is no longer necessary to manually adjust the ADSM settings each time drives are removed from ADSM because of technical fault or planned operations.

1.2.13.3 Single Drive Reclamation

For ADSM installations with only one media drive, the process of reclaiming media volumes is a manual task in ADSM Version 2. In ADSM Version 3, this process is automated by the server. By defining a reclaim storage pool for sequential access devices, the administrator can have ADSM automate the reclamation process



Event Logging, Reporting, and Monitoring

- Central logging of client events
- Client and Server event reporting
  - ► SNMP
  - ► File
  - ► User exit
  - ► Windows NT eventlog
  - ► Server-to-server
  - ► Tivoli
  - ► NetView for MVS
- SNMP monitoring of server

Copyright IBM Corporation 1997, 1998

1.2.14 Event Logging, Reporting, and Monitoring

With ADSM Version 3 events can be centrally logged, monitored, and reported by using industry-standard interfaces. Thus ADSM implementations can be integrated with system management applications and centralized control is facilitated.

1.2.14.1 Central Logging of Client Events

With ADSM Version 3 certain client messages can be logged as events on the ADSM server. Before ADSM Version 3, messages could be stored only in the client's error log and schedule log. To view those logs, the administrator had to have access to the client system. With ADSM Version 3 client messages can be collected to one central point. The intention of client message logging is to log problems encountered during an ADSM client operation. Therefore only messages indicating an error condition are logged as events. The only exception to this is client backup statistics, which also can be centrally logged.

For all events that are to be logged, each must be enabled by either message number or severity. Enabled client events that are logged to the ADSM server are, by default, stored in the activity log and displayed on the server console.

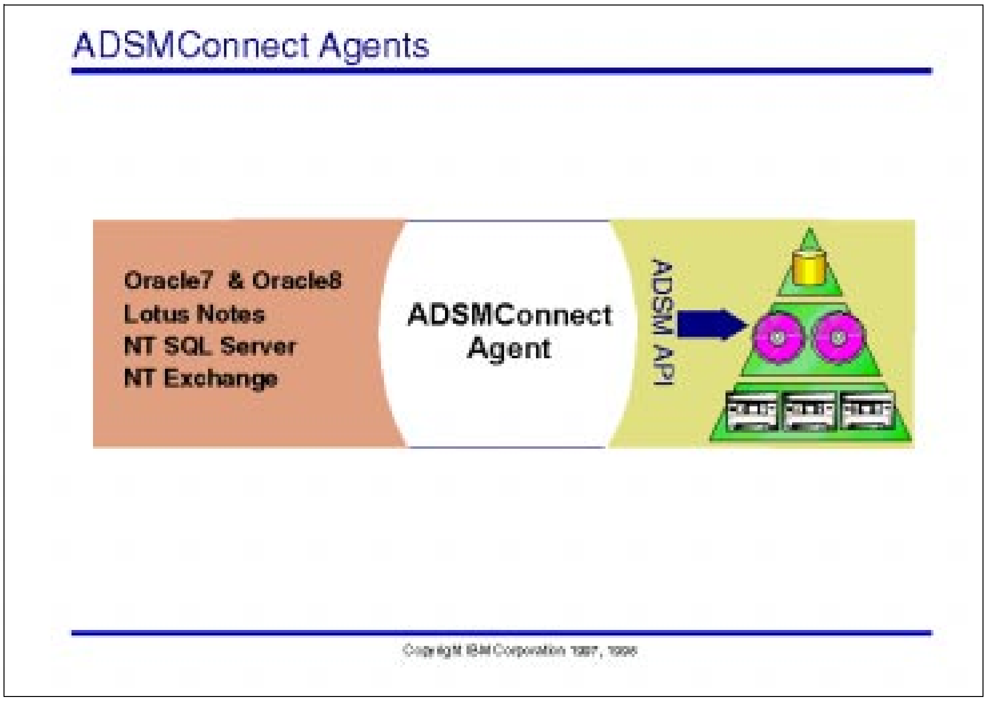
1.2.14.2 Client and Server Event Reporting

To take advantage of standard systems management interfaces, ADSM Version 3 provides the ability to send client and server events to external interfaces. Supported interfaces are SNMP managers such as NetView for AIX, CA Unicenter, or HP OpenView; Tivoli/Enterprise Console; NetView for MVS; a user exit; a file; another ADSM server using server-to-server communications; or the Windows NT Eventlog. Interfaces that receive ADSM event data are called *receivers*. Each event message, client or server, can be enabled for any of the supported receivers. It is possible to enable one message or a group of messages for more than only one receiver. As with client event logging, events are enabled for receivers by message number or severity.

The concept of redirecting event messages to external interfaces allows ADSM implementations to be integrated with systems management reporting tools.

1.2.14.3 SNMP Heartbeat Monitoring of Server

SNMP is also used to monitor network elements from a central point. It enables the monitored systems to send traps notifying the SNMP manager about events taking place on the local system. In addition to sending traps, a heartbeat monitor can be established to monitor whether managed ADSM servers are still alive. To enable ADSM to take advantage of SNMP monitoring, ADSM Version 3 includes an interface for SNMP. The interface is distributed with the server in the form of an SNMP subagent. It is supported for the ADSM server running on AIX and Windows NT. Communication between the ADSM server and the SNMP manager is established through these connection channels: ADSM server-SNMP subagent, SNMP subagent-SNMP agent, and SNMP agent-SNMP manager. To enable SNMP subagent-SNMP agent communication, the SNMP agent must support the Distributed Protocol Interface (DPI).



ADSMConnect Agents

Oracle7 & Oracle8
Lotus Notes
NT SQL Server
NT Exchange

ADSMConnect Agent

ADSM API

Copyright IBM Corporation 1997, 1998

1.2.15 ADSMConnect Agents

ADSM has provided integration with many popular applications:

- DB2 for AIX, HP/UX, SINIX, Solaris, OS/2, Windows NT
- Informix for AIX, HP/UX, Solaris
- Lotus Notes for OS/2
- OnDemand for AIX
- SAP with Oracle for AIX, Digital UNIX, HP/UX, NCR UNIX, SINIX, Solaris, Windows NT
- SAP with ADABAS-D for AIX, HP/UX, SINIX, Solaris, Windows NT

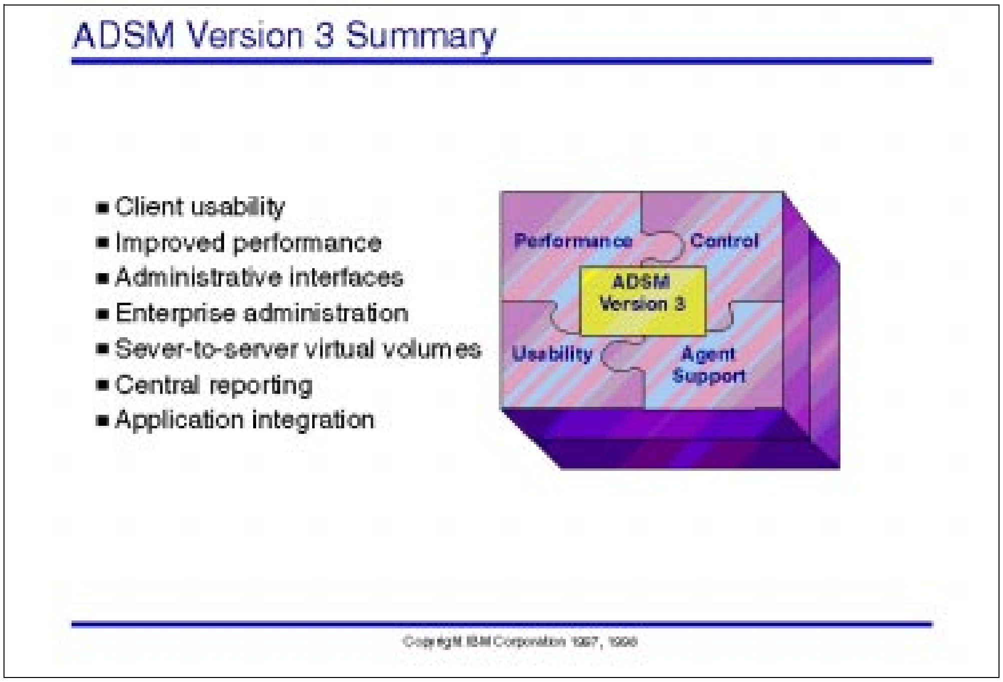
ADSM extends application integration by delivering new ADSMConnect Agents. The first agent, the ADSMConnect Agent for Oracle Backup on AIX, was delivered in March 1997. These new ADSMConnect Agents have become available during 1998:

- Oracle for Sun Solaris
- Lotus Notes for AIX
- Lotus Notes for Windows NT
- Microsoft SQL Server for Windows NT
- Microsoft Exchange for Windows NT

Note

These ADSMConnect Agents are not part of the ADSM Version 3 announcement. They are compatible with ADSM Version 2 and 3 servers.

The ADSMConnect agents are not covered further in this redbook.



ADSM Version 3 Summary

- Client usability
- Improved performance
- Administrative interfaces
- Enterprise administration
- Sever-to-server virtual volumes
- Central reporting
- Application integration

Performance

Control

ADSM Version 3

Usability

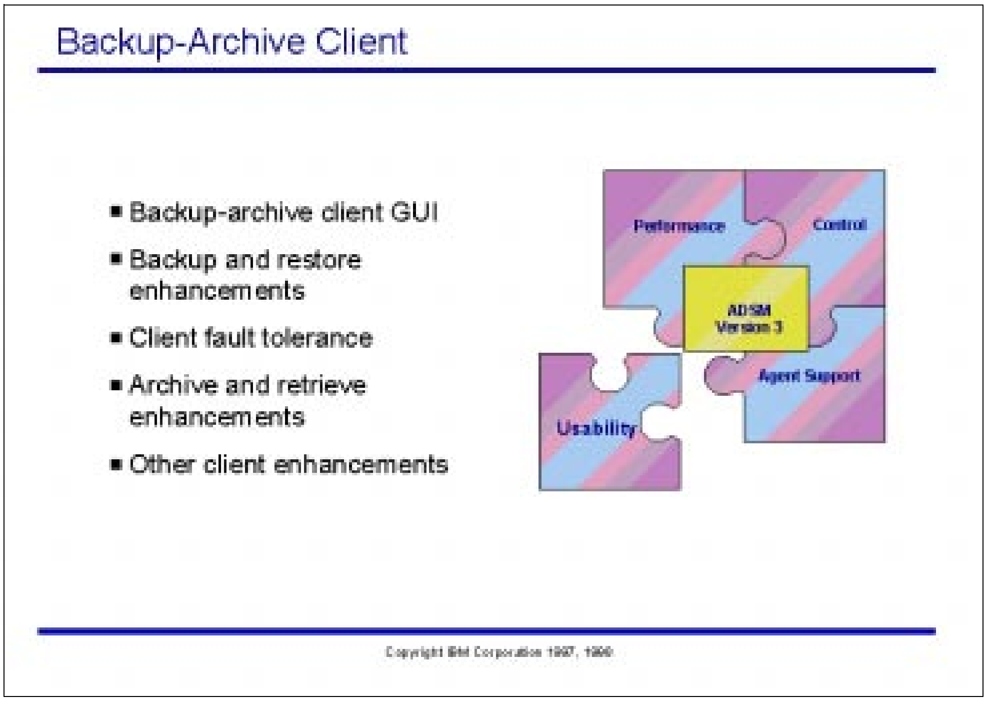
Agent Support

1.2.16 ADSM Version 3 Summary

ADSM Version 3 introduces these powerful new features designed to build on the success of ADSM Version 2 by meeting customers requirements for today and the future:

- Client usability through advanced storage management features and a user-friendly GUI
- Improved performance from intelligent algorithms, data transfer mechanisms, and fault tolerance
- Administration enhancements, including usability, a Web administrative interface, and new media management
- Enterprise administration to facilitate the administration of large-scale ADSM environments
- Better control and disaster protection through server-to-server virtual volumes
- Central reporting by integrating ADSM with systems management suites or by using SQL queries
- Application integration with industry-leading databases such as Oracle and Lotus Notes

Chapter 2. Backup-Archive Client



Backup-Archive Client

- Backup-archive client GUI
- Backup and restore enhancements
- Client fault tolerance
- Archive and retrieve enhancements
- Other client enhancements

Performance
Control
ADSM Version 3
Agent Support
Usability

Copyright IBM Corporation 1997, 1998

This chapter covers the Version 3 backup-archive client. The following topics are covered:

- Backup-archive client GUI

  Version 3 introduces a new GUI for the backup-archive client. This section discusses the look, feel, and usability of the new backup-archive client GUI.

- Backup and restore enhancements

  Several new functions have been added to the backup-archive client. This section covers the enhanced client functions such as No Query Restore and point-in-time restore.

- Client fault tolerance

  A major new usability feature in Version 3 is the enhanced client fault tolerance. This section cover these fault tolerant features.

- Archive and retrieve enhancements

  Version 3 introduces *Archive Package*s that extends the way the archive function is used. This section covers this and other archive enhancements.

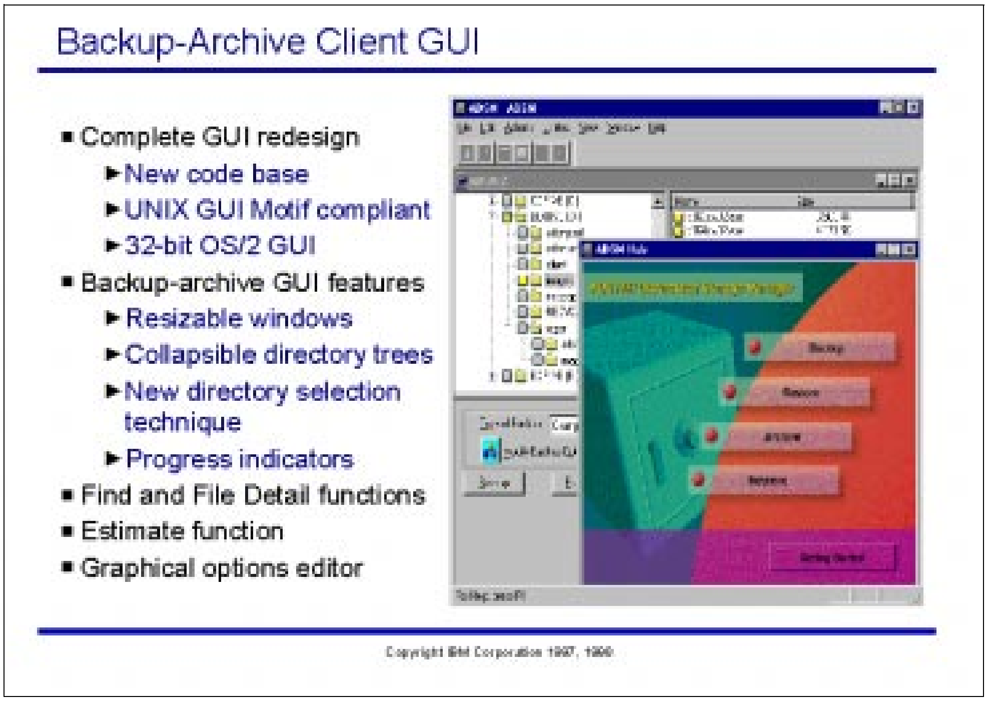
- Other client enhancements

  This section covers other new Version 3 client-related enhancements.

Web Backup-Archive Client

The Web backup-archive client introduced in ADSM Version 3.1.2 is covered in 7.3, “Web Backup-Archive Client” on page 226.

2.1 Backup-Archive Client GUI



Backup-Archive Client GUI

- Complete GUI redesign
  - New code base
  - UNIX GUI Motif compliant
  - 32-bit OS/2 GUI
- Backup-archive GUI features
  - Resizable windows
  - Collapsible directory trees
  - New directory selection technique
  - Progress indicators
- Find and File Detail functions
- Estimate function
- Graphical options editor

The Version 2 end-user interface was designed when clients had relatively few files and a small amount of data. Since then, tremendous growth has taken place in both the number of files and the amount of data on computers.

The graphic shows the new Version 3 Win32 backup-archive client GUI with the Hub window overlaying a Backup window. This section covers the following:

- Complete GUI redesign

  The backup-archive client GUI has been redesigned for Version 3. The new GUI offers an easy-to-use and intuitive way of performing functions such as backup or restore. File and directory selection is user friendly and efficient, enabling the backup-archive client to handle clients with filespaces containing a large number of files. The backup-archive client GUI uses a completely new code base. The UNIX backup-archive client interface has been coded with Motif libraries and provides a user interface with standard Motif control elements like scroll bars, buttons, and selection boxes. The OS/2 backup-archive client is coded with the 32-bit OS/2 APIs to increase performance and provide better memory management.

- Backup-archive client GUI features

  The Version 3 backup-archive client GUI uses resizable windows and collapsible directory trees. During a backup operation the backup-archive client GUI shows a tree structure representing directories and files on the client disk. During a restore operation it shows backed-up directory structures and backed-up files residing on the ADSM server. Files and directory structures are selected for backup and restore by clicking on, or expanding, directory icons. A progress indicator shows a user that some ADSM activity is occurring, for example, files are being scanned or backed up.

- Find and file detail functions

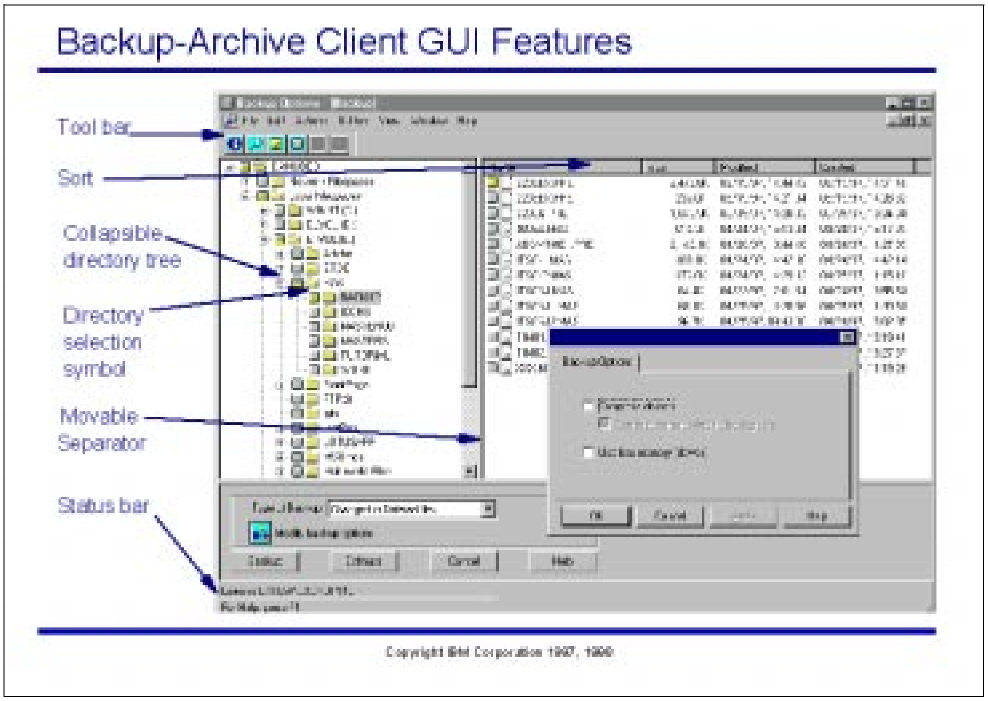
  New function has been added to the GUI to assist users in finding and identifying files for ADSM operations such as backup and restore.

- Estimate function

  The Version 3 backup-archive client GUI allows an Estimate function to be performed before the start of an ADSM operation. A user can choose to cancel the operation before it starts if the amount of data selected or the estimated elapsed time for the operation is excessive.

- Graphical options editor

  Changing the backup-archive client configuration files with a text editor is potentially an error-prone task. A graphical options editor has been added to the backup-archive client GUI to make customization of the client configuration options more user friendly and less prone to errors.


Backup-Archive Client GUI Features

Copyright IBM Corporation 1997, 1998

2.1.1 Backup-Archive Client GUI Features

When started, the backup-archive client GUI occupies only a small amount of space on the screen. It has resizable windows and scroll bars to allow users to resize or navigate windows. Several functions that are separate menu items in the Version 2 GUI are now integrated.

2.1.1.1 Tool Bar

A context-sensitive tool bar has been added to the new backup-archive client GUI to enhance usability. The tool bar contains icons used to perform common functions (such as search for, filter files, and display file details). The icons available on the tool bar depend on the ADSM function that is being performed.

2.1.1.2 Sort

Column headers function as sort buttons. For example, clicking on the Name column performs a sort by name. This sort function provides an easy way of identifying files in the backup-archive client GUI.

2.1.1.3 Collapsible Directory Tree

When an ADSM operation such as backup or restore is selected from the ADSM GUI, a window consisting of two parts is shown. The two parts include the Directory view (where a collapsible directory tree representing directories on the client disk is displayed) and the File view (where detailed information about files is displayed). The collapsible directory tree is initially shown with a client’s ADSM nodename as the top level (parent). A user can expand the nodename into filespaces (file system or disk partitions) and eventually directories by clicking on the expand (**+**) symbol in front of the directory (open the directory). The directory tree can be traversed recursively by expanding subdirectory branches in the directory tree. Parts of the directory structure can be collapsed and hidden from view by clicking on the collapse symbol (**-**) in front of the directories. Directories and files, selected for an ADSM operation such as backup or archive, are highlighted in the GUI.

The backup-archive client is designed to manage client filespaces containing a large number of files. Detailed file and directory information is not requested or displayed until needed. In a restore operation, the ADSM server does not send information to the client until the user expands a directory. In a backup operation, detailed directory information is not read from the client disk until the directory is expanded.

2.1.1.4 Directory Selection Symbol

Selecting directories of files for backup is made easy by providing a selection symbol in front of all directories and files displayed. Clicking on the selection symbol in front of a directory selects all files and subdirectories within that directory. Clicking on the selection symbol in front of a file name selects that individual file. Directories and files that have been selected have the selection symbol highlighted. Selected directories and files can be deselected by clicking on the selection symbol a second time.

Files excluded by include-exclude statements are highlighted in the GUI to provide easy verification of the include-exclude statements.

2.1.1.5 Movable Separator

Use of a movable separator enables a user to change the size of different sections of the backup-archive user interface.

2.1.1.6 Status Bar

The backup-archive client GUI displays messages informing a user that an operation is taking place. The GUI also contains graphical indicators that display progress during operations such as backup or restore.

Changes have been made to the final transfer statistics displayed by the GUI and the command line client at the end of an ADSM operation. The Version 3 backup-archive client calculates and displays two new transfer rates:

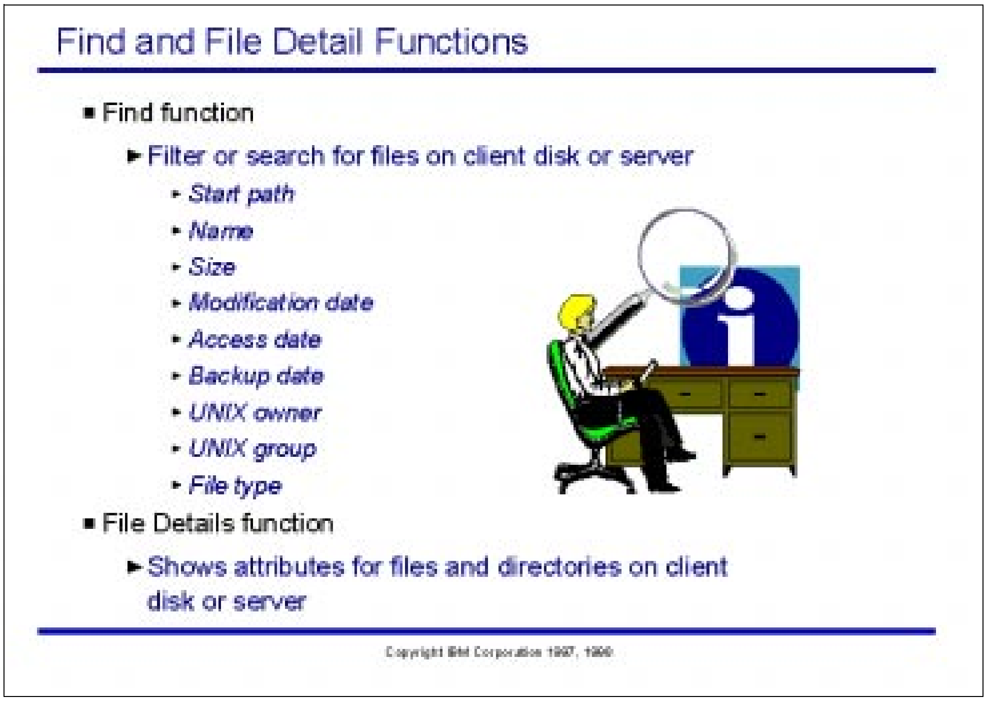
Network Data Transfer Rate This is calculated in the same way as the Version 2 data transfer rate was calculated (total number of bytes divided by the time the client spent sending data to the communication layer). The value indicates how efficient the network between the ADSM client and server is.

Aggregate Data Transfer Rate This is the average throughput and is calculated by dividing the number of bytes transferred by the elapsed time for the operation. This value includes any delays caused by tape mounts and other operations.

To further enhance usability, the dots used by the command line backup-archive client to display backup or restore progress for a file have been replaced by more meaningful progress indicators. The Version 3 backup-archive command line client displays the percentage of a file that has been backed up or restored:

```
dsmc> sel C:\adsmv3\*.exe
Selective Backup function invoked.

Directory-->                   0 C:\ADSMV3 <100%> [Sent]
Normal File-->         3,096,525 C:\ADSMV3\v381oos2.exe <35%> [ - ]
```

The UNIX backup-archive client GUI supports standard X Windows resources that allow users to change the behavior and appearance of the interface. The UNIX client X application resource file is located in the /usr/lib/X11/app-defaults directory.



Find and File Detail Functions

- Find function
  - Filter or search for files on client disk or server
    - *Start path*
    - *Name*
    - *Size*
    - *Modification date*
    - *Access date*
    - *Backup date*
    - *UNIX owner*
    - *UNIX group*
    - *File type*
- File Details function
  - Shows attributes for files and directories on client disk or server

Copyright IBM Corporation 1997, 1998

2.1.2 Find and File Detail Functions

To make the task of finding and identifying files and directories easy, two new functions have been added to the backup-archive client GUI.

2.1.2.1 Find Function

The Find function filters or searches for files stored locally on a client's disk or on the ADSM server. Invoking the Find function brings up a window to filter files or search for files matching certain criteria. The files matching the filter or find criteria can then be selected to perform an ADSM action (backup, archive, retrieve, restore, or delete archive file).

The Find function supports filter or search criteria such as start path, name, size, modification date, backup date. On multiuser UNIX platforms, owner, group, and file type can also be used for filtering or searching.

Start path The starting point for the search or filter operation. The start path can be a filespace (volume, partition, or file system mount point) or a directory.

Name A search or filter pattern using the following constructs:

- any name
- contains
- ends with
- is
- matches mask
- starts with

Size A size or filter pattern using the following constructs:

- exactly equal to
- larger or equal to

•smaller or equal to

Modification date Date or date range within which the file was modified

Access date Date or date range within which the file was last accessed

Backup date Date or date range within which the file was backed up (only from Restore window)

Owner The UNIX owner of the file (root user only)

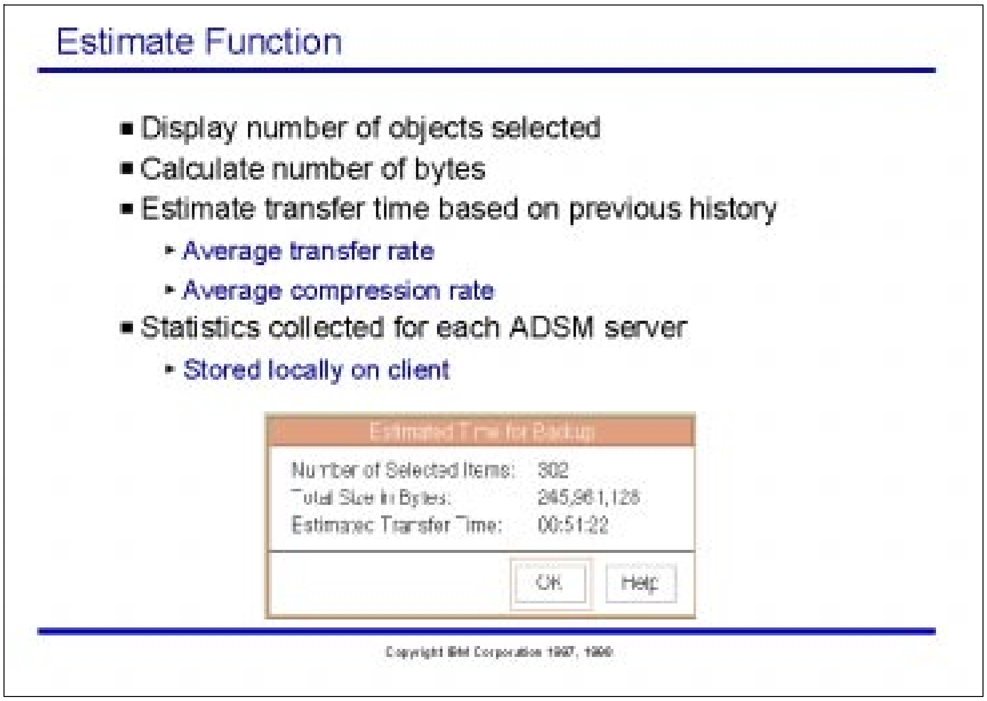
Group The UNIX group of the file (root user only)

File type The Version 3 UNIX backup-archive client supports backup and restore of special files such as block special files or character special files (UNIX only)

2.1.2.2 File Details Function

The File Details function displays detailed information about a file or directory that is managed by ADSM or local on a client disk. File Details shows detailed information about the file or directory that is currently selected:

- Name
- Size
- Type (UNIX only)
  - File
  - Directory
  - Filespace
- Userid (UNIX only)
- Groupid (UNIX only)
- Modification date
- Creation date
- Last access date
- Attributes (archive bit, extended attributes)



Estimate Function

- Display number of objects selected
- Calculate number of bytes
- Estimate transfer time based on previous history
  - Average transfer rate
  - Average compression rate
- Statistics collected for each ADSM server
  - Stored locally on client

Estimated Time for Backup

Number of Selected Items: 302
Total Size in Bytes: 245,961,128
Estimated Transfer Time: 00:51:22

Copyright IBM Corporation 1997, 1998

2.1.3 Estimate Function

The Backup, Restore, Archive, and Retrieve windows have an Estimate button that is used to display the amount of data selected and the estimated time it will take to perform the ADSM operation. Clicking on this button displays an Estimate window with a variety of information.

2.1.3.1 Number of Objects Selected

The Estimate function displays the number of objects (files and directories) selected for an ADSM operation such as backup or restore.

2.1.3.2 Calculated Size

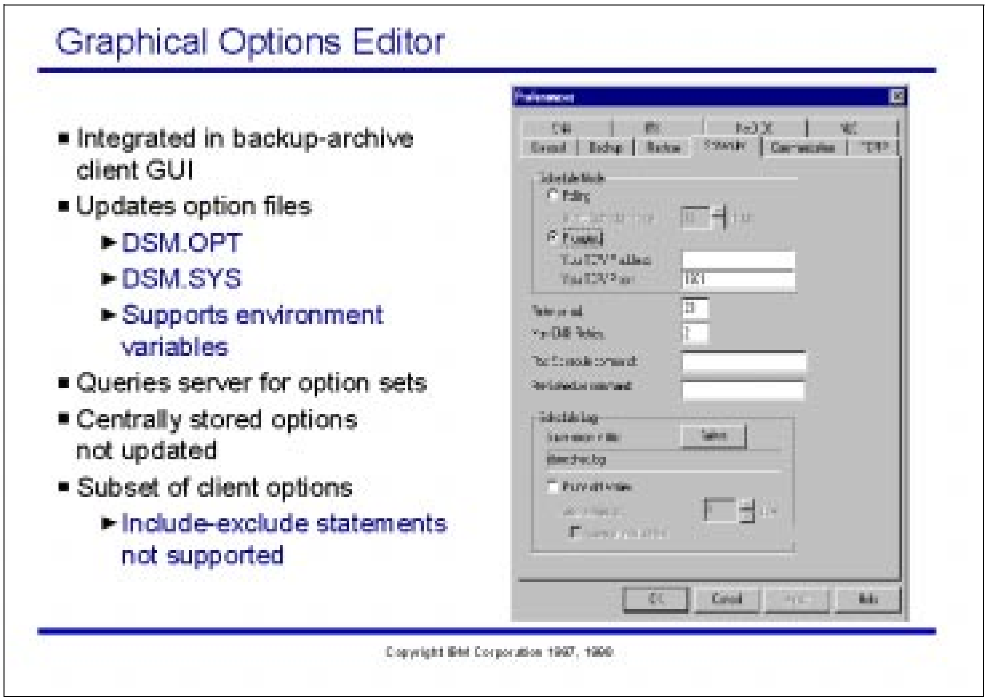
The Estimate function calculates the number of bytes the currently selected objects occupy by scanning the selected directories or requesting file information from the ADSM server.

2.1.3.3 Estimated Transfer Time

The ADSM backup-archive client estimates the elapsed time for the operation on the basis of historical information. The estimation is calculated by using the average transfer rate and average compression rate from previous operations.

2.1.3.4 Statistics from ADSM Server

The historical data is stored on disk on the client. If the client connects to multiple servers, this information is recorded on a per server basis. For UNIX backup-archive clients, this information is stored in a file named .adsmrc in the user’s home directory. For PC clients, this file is dsm.ini in the backup-archive client directory.



Graphical Options Editor

- Integrated in backup-archive client GUI
- Updates option files
  - DSM.OPT
  - DSM.SYS
  - Supports environment variables
- Queries server for option sets
- Centrally stored options not updated
- Subset of client options
  - Include-exclude statements not supported

2.1.4 Graphical Options Editor

A new graphical options editor is provided with the Version 3 backup-archive client GUI. The graphical options editor makes the process of updating the ADSM client options more user friendly and less error prone than before.

2.1.4.1 Integrated in Backup-Archive Client GUI

The graphical options editor is integrated in the backup-archive client GUI on all client platforms with a GUI and is started by selecting Preferences from the Edit pull-down menu.

2.1.4.2 Updates Options Files

The graphical options editor updates dsm.opt and dsm.sys (UNIX clients only) if any options are changed. The client configuration files can still be updated with a text editor on the client.

The options editor uses the ADSM environment variables, DSM_DIR and DSM_CONFIG, to locate the client configuration files.

2.1.4.3 Queries Server for Option Sets

The graphical options editor queries the server for client options that are stored centrally in client option sets on the server. The centrally stored options are displayed by the graphical options editor. Only client options that the backup-archive client is authorized to update can be changed with the graphical options editor.

2.1.4.4 Centrally Stored Options Not Updated

The graphical options editor only updates or adds client options to the option file(s) on the client. It does not update client option sets on the ADSM server.

2.1.4.5 Subset of Client Options

The graphical options editor does not currently support updates to include-exclude statements. The options editor groups the options in the following categories:

General General options such as nodename

Backup Options that have to do with backup, for example, compression and number of retries

Restore File replace options

Scheduler Options that influence scheduling services, such as schedule mode and schedule log file handling

Communication The communication method to use and its timeout parameters

TCPIP TCP/IP definitions, such as ADSM server name, TCP window, and buffer sizes

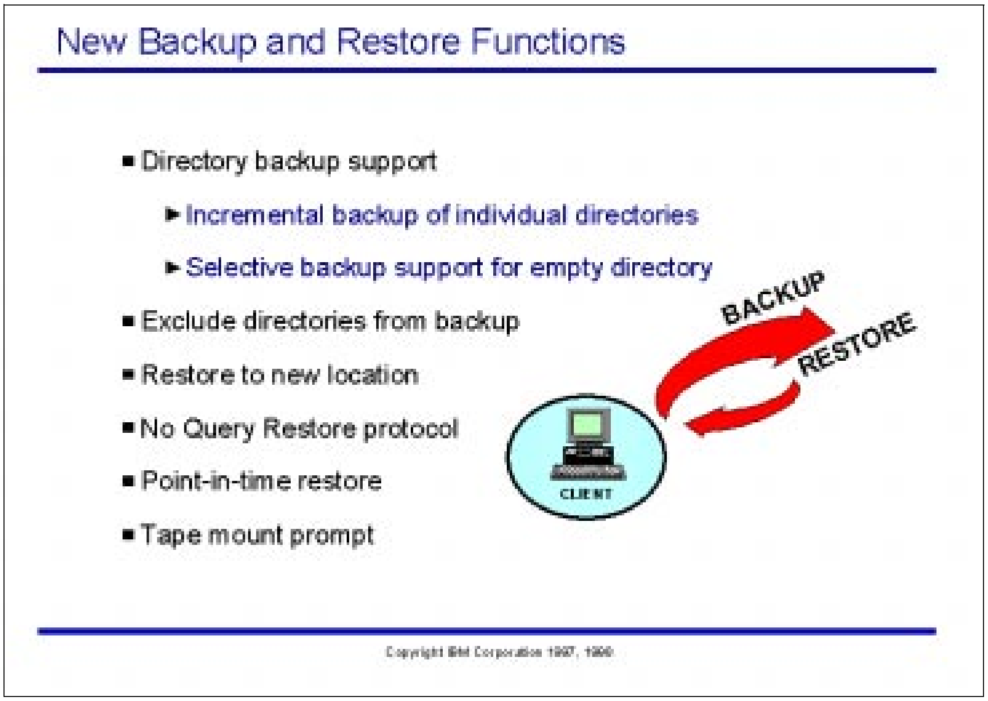
SNA SNA definitions

IPX IPX definitions

NetBIOS NetBIOS definitions

NLS Language, date, time, and number formats.

2.2 New Backup and Restore Functions



New Backup and Restore Functions

- Directory backup support
  - Incremental backup of individual directories
  - Selective backup support for empty directory
- Exclude directories from backup
- Restore to new location
- No Query Restore protocol
- Point-in-time restore
- Tape mount prompt

BACKUP
RESTORE
CLIENT

Copyright IBM Corporation 1997, 1998

New function has been added to the backup-archive client to enhance ease of use and provide enhanced restore function. This section covers the following topics:

- Directory backup support

  The directory support in Version 3 has been enhanced. The backup-archive client now supports incremental backup of subdirectories, whereas the Version 2 backup-archive client supports only incremental backup at the client filespace level. During a selective backup, the ADSM backup-archive client backs up empty directories and their attributes. New command line options allow a user to choose whether the files, the directories, or both should be backed up or restored.

- Exclude directories from backup

  Although directories can be excluded from backup in Version 2, only the files are excluded. Excluded directories are scanned, and the directory objects are backed up. The Version 3 backup-archive client supports exclude of directory structures from backup.

- Restore to new location

  The backup-archive client supports restore of directory structures to new locations while preserving the directory structure.

- No Query Restore protocol

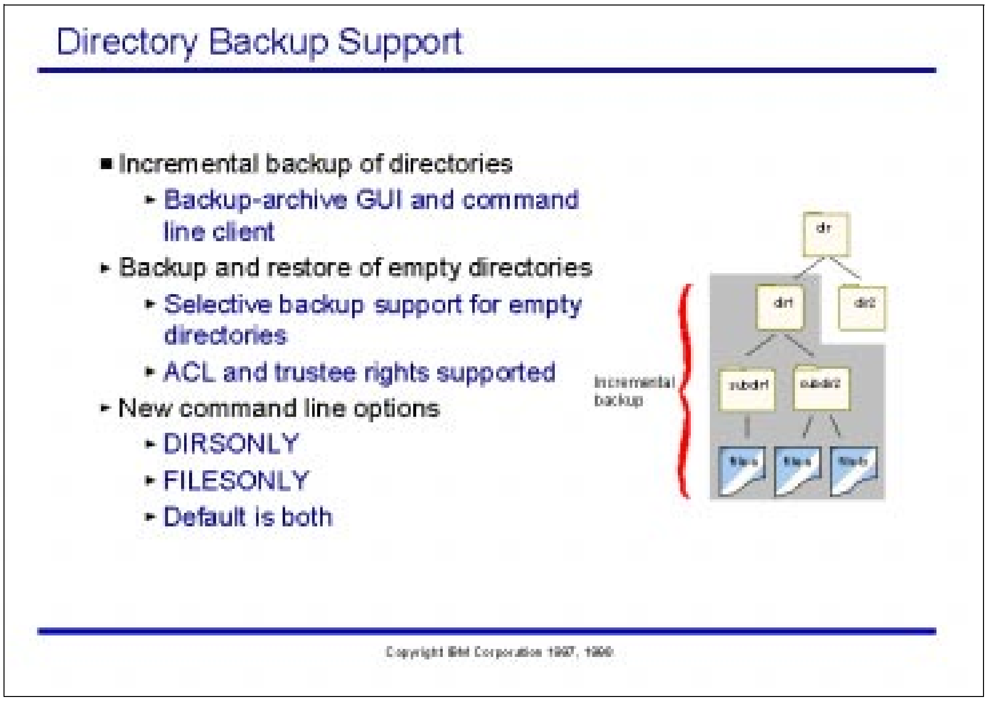
  No Query Restore is a new protocol that is used between Version 3 servers and clients. Many of the new Version 3 restore enhancements use the protocol.

- Point-in-time restore

In Version 2 it is difficult to perform a restore of a complete filespace or a directory to the state it had prior to a given date and time. Version 3 supports point-in-time restore.

- Tape mount prompt

  When performing restores using the No Query Restore protocol, a user is presented with new cancel and skip file options when data is on tape.



Directory Backup Support

- Incremental backup of directories
  - Backup-archive GUI and command line client
- Backup and restore of empty directories
  - Selective backup support for empty directories
  - ACL and trustee rights supported
- New command line options
  - DIRSONLY
  - FILESONLY
  - Default is both

incremental backup
dir
dir1
dir2
subdir1
subdir2

Copyright IBM Corporation 1997, 1998

2.2.1 Directory Backup Support

The Version 3 backup-archive client is enhanced to support backup and restore of directories in a more intuitive and logical way.

2.2.1.1 Incremental Backup of Directories

In Version 2 an incremental backup is performed at the filespace level (file system, partition, or volume). The Version 3 backup-archive client supports incremental backup of directories. Incremental backup on the directory level can greatly reduce the amount of data that is backed up and the time it takes to perform the backup.

The new support for incremental backup of directory structures is invoked automatically if a directory is selected from the ADSM backup-archive client GUI or in the command line client if an incremental command specifying a directory is used:

```
dsmc> incremental /home/tim
```

2.2.1.2 Backup and Restore of Empty Directories

Version 2 only backs up empty directory structures when an incremental backup is performed from the backup-archive command line client. A selective backup or an incremental backup from the backup-archive client GUI does not back up empty directories and directory attributes. This may cause problems for customers that have clients with large and empty directory structures.

The Version 3 backup-archive client selective backup has been enhanced to support backup and restore of empty directory structures. All directory attributes such as ACL, trustee rights, and file permissions are supported.

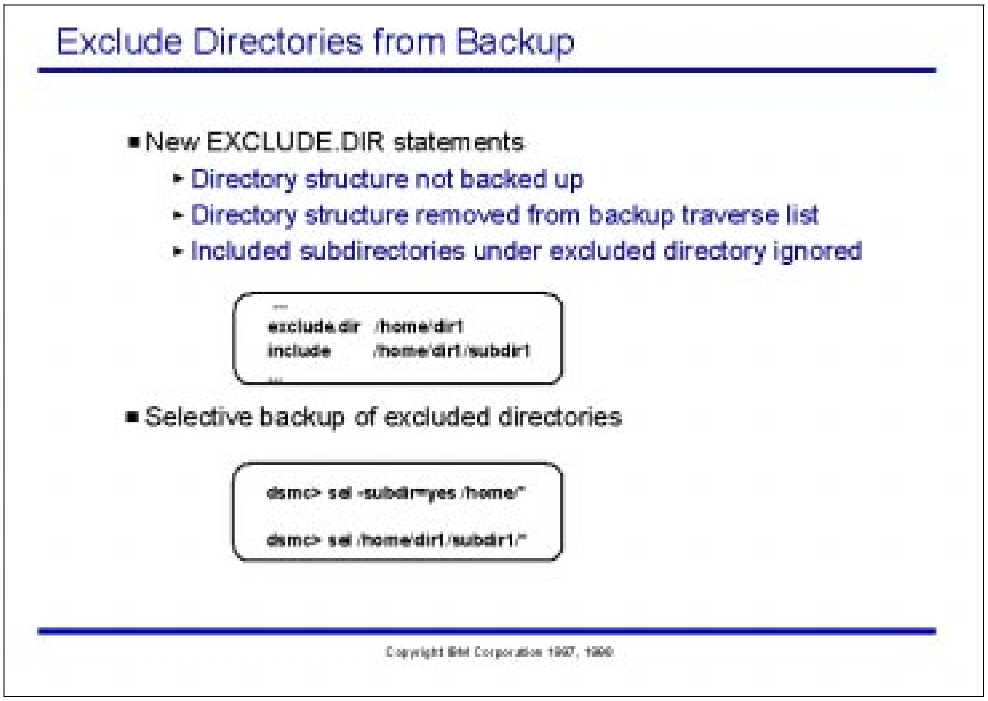
2.2.1.3 New Command Line Options

New command line options have been added to make it possible to select only directories or files for a backup, restore, or query operation:

DIRSONLY When DIRSONLY is specified, only the directories and their attributes are backed up or restored. This option allows a user to back up a directory tree without backing up the files or to force a backup of directory attributes such as ACL, trustee rights, and extended attributes (EAs).

FILESONLY When FILESONLY is specified, only the files and their attributes are backed up or restored.

If none of the options is used, both directories and files are backed up or restored. The DIRSONLY and FILESONLY options can be specified only on backup or restore commands using the backup-archive command line client. The options cannot be put in the client user options file (dsm.opt).



Exclude Directories from Backup

- New EXCLUDE.DIR statements
  - Directory structure not backed up
  - Directory structure removed from backup traverse list
  - Included subdirectories under excluded directory ignored

```
...
exclude.dir   /home/dir1
include       /home/dir1/subdir1
...
```

- Selective backup of excluded directories

```
dsmc> sel -subdir=yes /home/*

dsmc> sel /home/dir1/subdir1/*
```

2.2.2 Exclude Directories from Backup

When a directory structure is excluded from backup in Version 2, the directory objects of the excluded directory structure are backed up. Only the files in an excluded directory structure are excluded from backup. Directories excluded through an EXCLUDE statement are nevertheless traversed by the ADSM backup-archive client during an incremental backup. Processing large directory structures can be time and resource consuming and can cause timeout problems on the clients.

The Version 3 backup-archive client supports excluding directory structures from backup.

2.2.2.1 New EXCLUDE.DIR Statements

A new include-exclude statement, EXCLUDE.DIR, is introduced in Version 3. The EXCLUDE.DIR statement excludes a directory structure from the internal traverse tree that the ADSM backup-archive client builds internally before performing the backup and prevents directories and directory attributes from being backed up.

The exclude statement goes in the client user options file (dsm.opt) on single-user clients like OS/2 and Windows NT, and in the include/exclude file referred to by the client system options file (dsm.sys) on multiuser operating systems like UNIX.

The new EXCLUDE.DIR statement coexists with the existing EXCLUDE statement:

EXCLUDE.DIR Excludes a directory structure from backup and from being traversed during incremental backup

EXCLUDE.FILE Can be abbreviated to EXCLUDE and excludes files from backup. Excluded directory structures are traversed during incremental backup.

If a directory structure is excluded through EXCLUDE.DIR, subdirectories in the excluded directory tree are not eligible for backup. Any INCLUDE statement that includes part of an excluded directory structure is ignored at backup time:

```
exclude.dir    /home/dir1
include        /home/dir1/subdir1
```

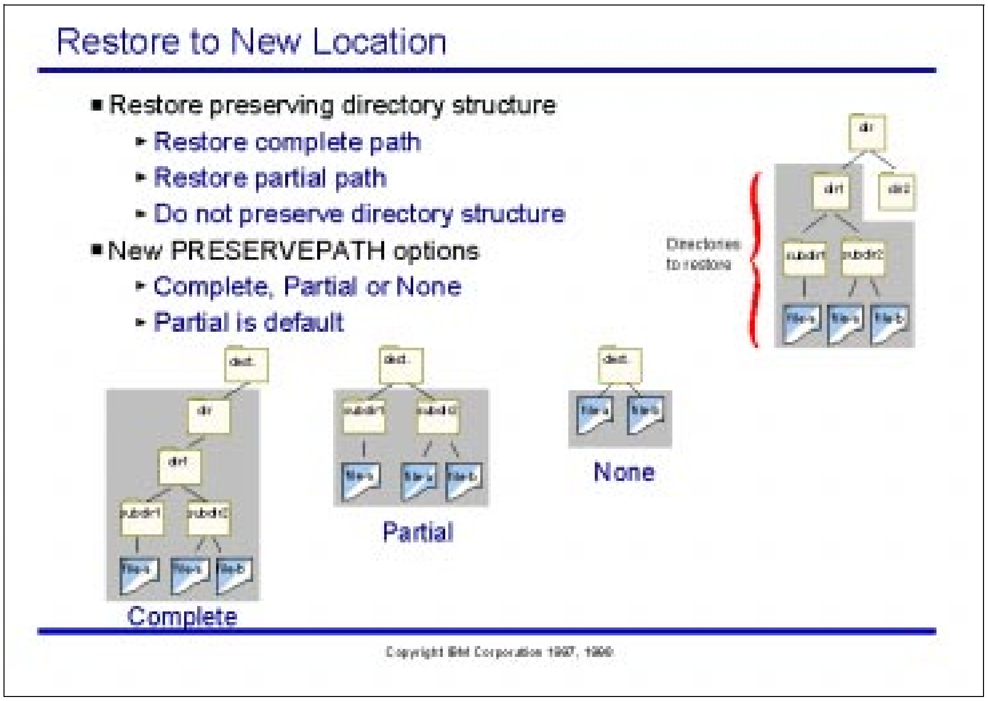
In the above example the include statement for the /home/dir1/subdir1 directory is ignored when an incremental backup is performed because the subdirectory has been excluded through an EXCLUDE.DIR statement.

2.2.2.2 Selective Backup of Excluded Directories

Even though a directory structure is excluded through the new EXCLUDE.DIR statement, subdirectories and files within the excluded directory structure can be backed up using the SELECTIVE backup command:

```
dsmc> selective  -subdir=yes /home/*

dsmc> selective  /home/dir1/subdir1/*
```

The first selective backup command backs up all files except those in the excluded directory (/home/dir1) because the EXCLUDE.DIR statement is honored for subdirectories. The second selective backup command backs up files in the /home/dir1/subdir1. This is an explicit backup of an excluded directory and overrides the EXCLUDE.DIR statement.



Restore to New Location

- Restore preserving directory structure
  - Restore complete path
  - Restore partial path
  - Do not preserve directory structure
- New PRESERVEPATH options
  - Complete, Partial or None
  - Partial is default

2.2.3 Restore to New Location

Preserving directory structures when restoring files to a new location is difficult with Version 2. This problem has been addressed in the Version 3 backup-archive client.

2.2.3.1 Restore Preserving Directory Structure

The Version 3 backup-archive client GUI presents three options when a restore or retrieve operation is invoked:

- Restore complete path

  The complete path starting with the root directory of the filespace (excluding filespace name) is restored to the new location (absolute path).

- Restore partial path

  The directory structures starting with the selected directory are restored to the new location (relative path).

- Do not preserve directory structure

  This option removes all directories and restores or retrieves all selected files to the new location. All files will be located in the same directory (flat restore).

2.2.3.2 New PRESERVEPATH Options

The PRESERVEPATH option has been added to the RESTORE command for the command line backup-archive client. This option allows users to specify how directory structures are handled when performing restore to a new location. In the following examples of using this option, the directory structure that has been backed up is:

```
C:\dir\dir1\subdir1\file-a
          \subdir2\file-a
          \subdir2\file-b
```

The PRESERVEPATH option can be specified with one of three options:

1. Complete

   The complete directory structure, all the way from the root directory (excluding filespace name), is created at the new location:

   ```
   dsmc> restore -subdir=yes -preservepath=complete c:\dir\dir1\* C:\tmp\

   Restoring           0 C:\DIR\dir1\subdir1 --> C:\tmp\DIR\dir1\subdir1<100%> Done
   Restoring           0 C:\DIR\dir1\subdir2 --> C:\tmp\DIR\dir1\subdir2<100%> Done
   Restoring          15 C:\DIR\dir1\subdir1\file-a --> C:\tmp\DIR\dir1\subdir1\file-a <100%> Done
   Restoring          15 C:\DIR\dir1\subdir2\file-a --> C:\tmp\DIR\dir1\subdir2\file-a <100%> Done
   Restoring          17 C:\DIR\dir1\subdir2\file-b --> C:\tmp\DIR\dir1\subdir2\file-b <100%> Done
   ```

2. Partial

   All subdirectories are created at the new location:

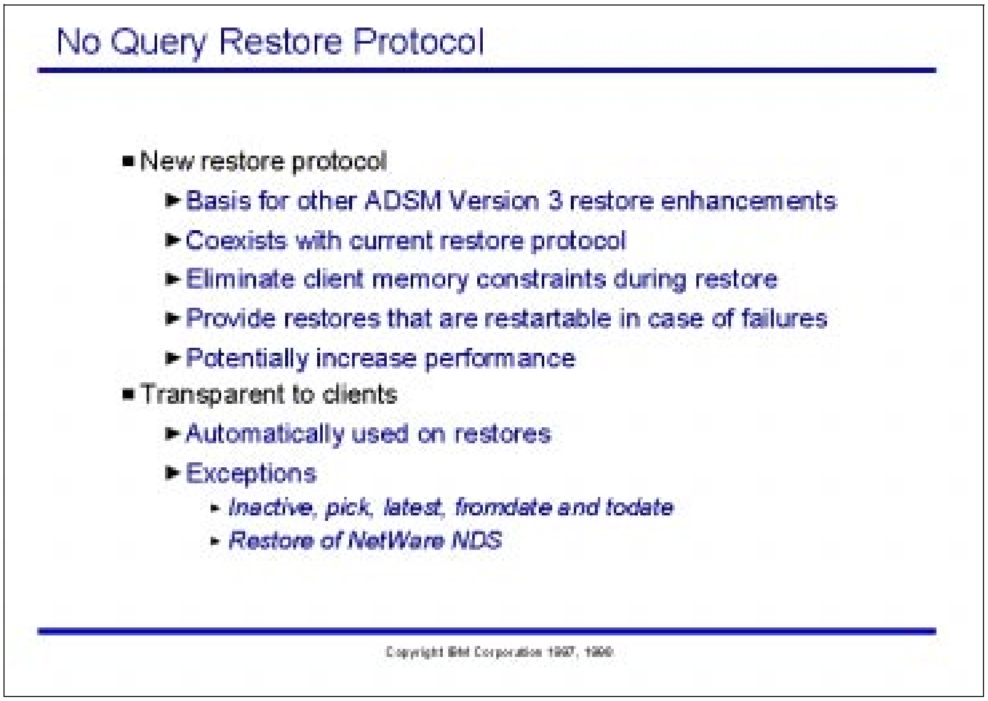
   ```
   dsmc> restore -subdir=yes -preservepath=partial c:\dir\dir1\* C:\tmp\

   Restoring           0 C:\DIR\dir1\subdir1 --> C:\tmp\dir1\subdir1 <100%> Done
   Restoring           0 C:\DIR\dir1\subdir2 --> C:\tmp\dir1\subdir2 <100%> Done
   Restoring          15 C:\DIR\dir1\subdir1\file-a --> C:\tmp\dir1\subdir1\file-a <100%> Done
   Restoring          15 C:\DIR\dir1\subdir2\file-a --> C:\tmp\dir1\subdir2\file-a <100%> Done
   Restoring          17 C:\DIR\dir1\subdir2\file-b --> C:\tmp\dir1\subdir2\file-b <100%> Done
   ```

3. None

   None of the directory structures is created at the new location. All files are restored "flat" in the same directory. Files with the same name located in different subdirectories are restored to the same directory at the new location. When the restore ends, only the last of the files with identical file names remain:

   ```
   dsmc> restore -subdir=yes -preservepath=none c:\dir\dir1\* C:\tmp\
   Restoring              15 C:\DIR\dir1\subdir1\file-a --> C:\tmp\file-a <100%> Done

   File C:\tmp\file-a exists, do you want to replace it?
   (1)YES - (2)NO - (3)YES ALL - (4)NO ALL - (A)ABORT
    which option  1,2,3,4,A ? 1
   Restoring          15 C:\DIR\dir1\subdir2\file-a --> C:\tmp\file-a <100%> Done
   Restoring          17 C:\DIR\dir1\subdir2\file-b --> C:\tmp\file-b <100%> Done
   ```

Partial is the default option.



No Query Restore Protocol

- New restore protocol
  - Basis for other ADSM Version 3 restore enhancements
  - Coexists with current restore protocol
  - Eliminate client memory constraints during restore
  - Provide restores that are restartable in case of failures
  - Potentially increase performance
- Transparent to clients
  - Automatically used on restores
  - Exceptions
    - *Inactive, pick, latest, fromdate and todate*
    - *Restore of NetWare NDS*

2.2.4 No Query Restore Protocol

Before Version 3, ADSM used one restore protocol, which required the server to send a list of backed up files to the client whenever a restore was performed. The list of files received from the server was sorted in the memory space on the client. The client then signaled the server with a list of files to be restored. The memory required on the client to hold and sort the list of files could use all available memory on memory-constrained clients, causing the restore to fail.

2.2.4.1 New Restore Protocol

No Query Restore is used by new client restore functions such as point-in-time restore and restores. It coexists with current restore protocols and is supported on all Version 3 servers and clients.

The No Query Restore protocol is provided to address potential memory constraints on ADSM clients during restore. The current protocol requires the ADSM client to query the ADSM server for a list of files backed up from a filespace (hence the phrase No Query Restore) and depends on the backup-archive client to perform a sort in memory of the list of files returned. For clients with large filespaces, the list of backed up files can become very large; in fact, too large to keep and sort in memory for memory-constrained clients. This is especially true for ADSM clients such as Novell NetWare where the client operating system does not provide virtual memory.

Restores using the No Query Restore protocol are restartable. Thus they can be restarted from the point of failure after situations such as network failures and server outages.

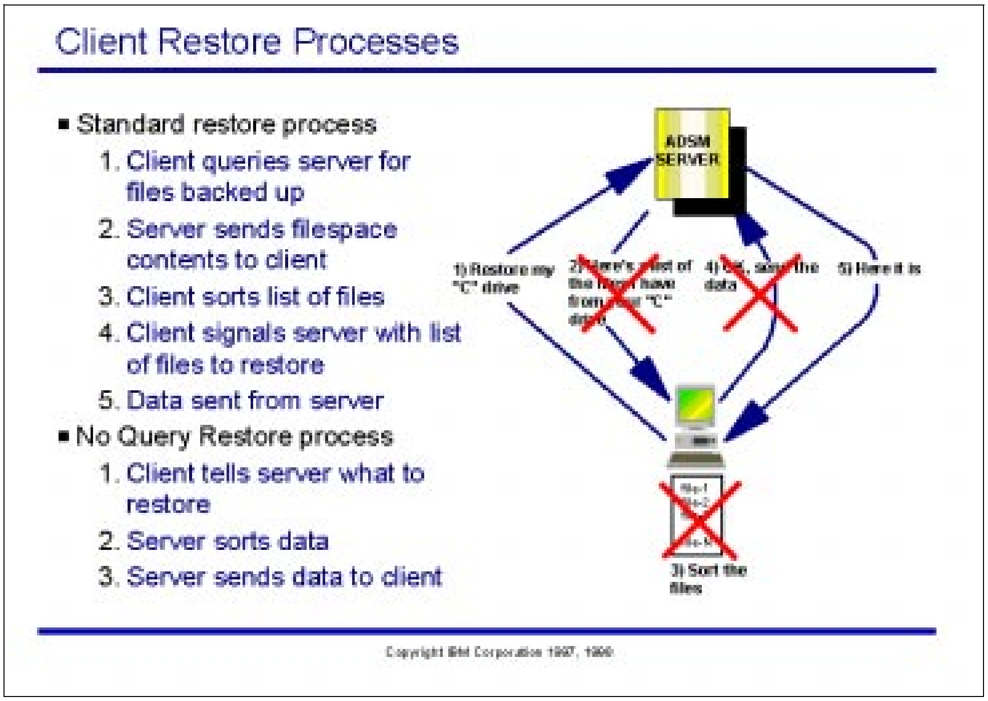
The No Query Restore moves some of the restore processing to the ADSM server, which can potentially increase restore performance on less-powerful clients.

2.2.4.2 Transparent to Clients

The use of the No Query Restore protocol is transparent to the ADSM client. The protocol is used automatically when a restore is performed. The only exceptions are restores using these options:

- inactive
- pick
- latest
- fromdate and fromtime
- todate and totime

Use of the above restore options results in file filtering and sorting being performed on the client. Also, for a restore of the NetWare Directory Services (NDS), the No Query Restore protocol is not used.



Client Restore Processes

- Standard restore process
  1. Client queries server for files backed up
  2. Server sends filespace contents to client
  3. Client sorts list of files
  4. Client signals server with list of files to restore
  5. Data sent from server
- No Query Restore process
  1. Client tells server what to restore
  2. Server sorts data
  3. Server sends data to client

Copyright IBM Corporation 1997, 1998

2.2.5 Client Restore Processes

The differences between the standard restore process and the No Query Restore process are described below.

2.2.5.1 Standard Restore Process

Whenever the standard restore protocol is used during a restore, the following steps are performed:

1. Client queries server for files backed up.

   The client queries the server for a list of files backed up for the client filespace being restored.

2. Server send filespace list to client.

   The server sends a list of backed up files that match the restore criteria. If both active and inactive files are to be restored, information about all backed-up files, both active and inactive, is sent to the client.

3. Client sorts list of files.

   The list of files returned from the server is sorted in client memory to determine the file restore order and to minimize tape mounts required to perform the restore.

4. Client signals server.

   The client tells the server to restore file data and directory objects.

5. Files sent from server.

   The directories and files to be restored are sent from the server to the client.

2.2.5.2 No Query Restore Process

The No Query Restore protocol is much simpler:

1. The client tells the server what to restore.

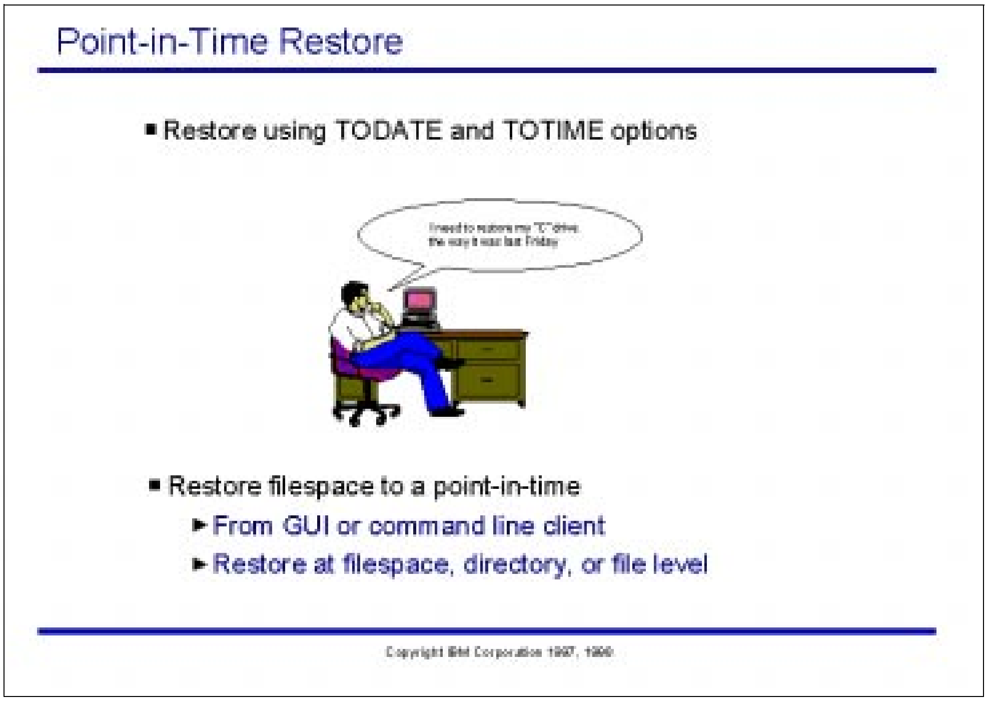
   The client informs the server that a No Query Restore is going to be performed and provides the server with the filespace and/or directory and file details.

2. Server sorts data.

   Using an internal sort table, the server performs the sorting necessary to minimize tape mounts.

3. Server sends data to client.

   The data to be restored is sent to the client. File and directory objects stored on disk are sent immediately because sorting is not required before the data is restored.



Point-in-Time Restore

- Restore using TODATE and TOTIME options

I need to restore my "C" drive
the way it was last Friday

- Restore filespace to a point-in-time
  - From GUI or command line client
  - Restore at filespace, directory, or file level

Copyright IBM Corporation 1997, 1998

2.2.6 Point-in-Time Restore

It is difficult to do a point-in-time restore with the RESTORE command and the TODATE and TOTIME options. Version 3 provides a new point-in-time restore function that can recover a filespace or a directory to the state it had at a prior date and time.

2.2.6.1 Restore Using TODATE and TOTIME Options

Using the existing TODATE and TOTIME restore options can result in situations where a true point-in-time restore is not performed:

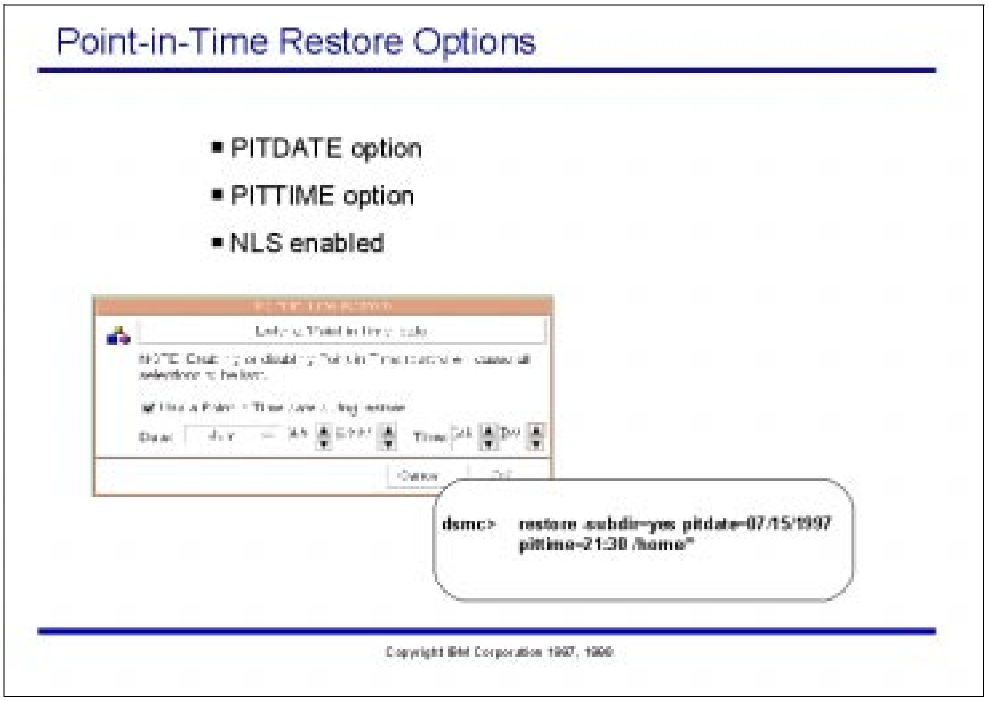
- Files deleted from the client filespace before the date and time specified using the TODATE and TOTIME options can be restored. These files, if restored, must be manually deleted after the restore operation.
- Restoring deleted files can potentially result in overcommitment of disk space on the client and situations where a restore fails because of lack of space.

2.2.6.2 Restore Filespace to a Point-in-Time

The Version 3 point-in-time restore function restores a filespace (or directory) to the state it had prior to a given date and time (actually, the date and time of the last incremental backup before the specified date and time). This support for point-in-time restore is essential for recovering a filespace or directory to a time when it was known to be in a good or consistent state. For example, a point-in-time restore can eliminate the effect of data corruption or recover a configuration to a previous date or time.

Both the Version 3 backup-archive client GUI and command line client support point-in-time restore when used with a Version 3 server. A point-in-time restore is selected from the backup-archive client GUI by selecting a Point-in-Time date from the Restore window. Point-in-time restores are started from the ADSM backup-archive command line client by using new command line options.

A point-in-time restore is supported on the filespace, directory, or file level. When a point-in-time restore is performed, new files that have been created on the client after the point-in-time date are not deleted.



Point-in-Time Restore Options

- PITDATE option
- PITTIME option
- NLS enabled

```
dsmc>   restore -subdir=yes pitdate=07/15/1997
        pittime=21:30 /home/*
```

Copyright IBM Corporation 1997, 1998

2.2.7 Point-in-Time Restore Options

The point-in-time restore function introduces new backup-archive client options. A point-in-time restore date and time can be specified from the backup-archive client GUI or from the backup-archive command line client, using the new PITDATE and PITTIME options.

2.2.7.1 PITDATE Option

The date for the point-in-time restore. Files and directories that existed on the client filespace when the last incremental backup was run before this date are candidates for restore.

2.2.7.2 PITTIME Option

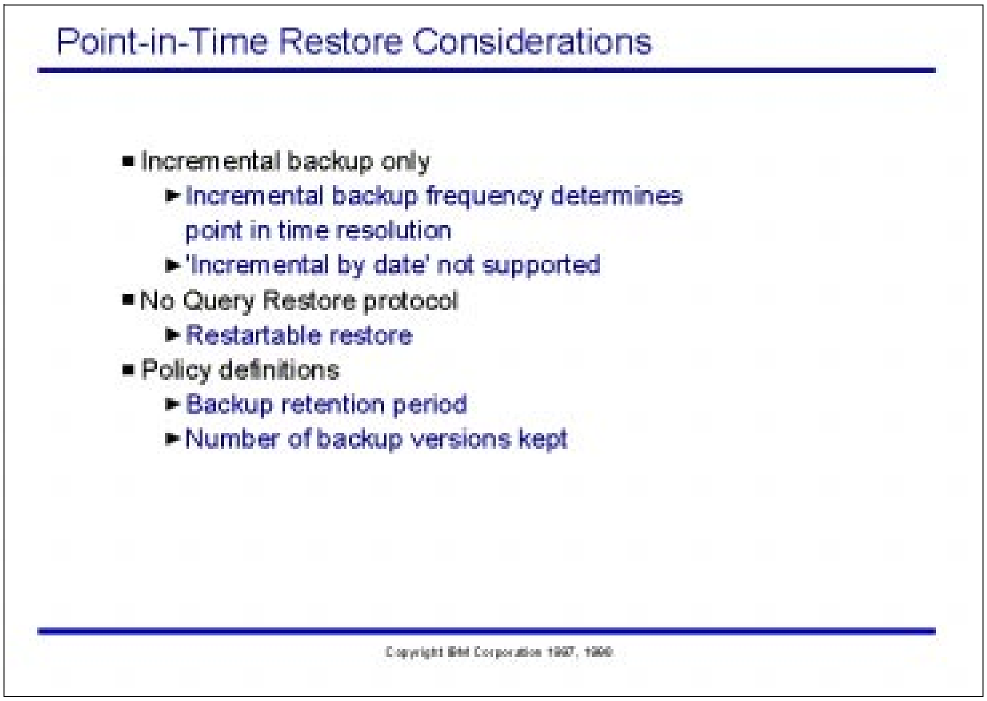
The time for the point-in-time restore. Files and directories that existed on the client filespace when the last incremental backup was run before this time are candidates for restore.

2.2.7.3 NLS Enabled

The PITDATE and PITTIME options are NLS enabled and use the date and time format specified in the client user options file (dsm.opt).

To guarantee a consistent restore, care should be taken when specifying a point-in-time date and time to ensure that they time do not overlap a time when incremental backups were run.

The example on the graphic shows a point-in-time restore that will restore the /home filesystem to the state it had before, or at, 21:30 on July 15. The example also shows how this point-in-time date is specified in the backup-archive client GUI.



Point-in-Time Restore Considerations

- Incremental backup only
  - Incremental backup frequency determines point in time resolution
  - 'Incremental by date' not supported
- No Query Restore protocol
  - Restartable restore
- Policy definitions
  - Backup retention period
  - Number of backup versions kept

Copyright IBM Corporation 1997, 1998

2.2.8 Point-In-Time Restore Considerations

Use of point-in-time restore has a number of dependencies and considerations.

2.2.8.1 Incremental Backup Only

Point-in-time restores are possible only when incremental backups are run on the client because the server is only notified about files that are deleted from a client filespace during an incremental backup. Incremental backups should run frequently enough to provide the necessary point-in-time resolution. Files that have been deleted from a client filespace between two incremental backups might be restored during a point-in-time restore.

Incremental backups performed with the *Incremental by date* method are not true incremental backups. A point-in-time restore cannot be performed when this type of incremental backup is used.

2.2.8.2 No Query Restore Protocol

Whenever a point-in-time restore is performed, the new Version 3 No Query Restore protocol is used. The use of the No Query Restore protocol means that a point-in-time restore will use other Version 3 restore enhancements, such as restartable restores, that build on the No Query Restore protocol.

During a point-in-time restore, the ADSM server does the file filtering, as opposed to other restores where the ADSM client does all the file filtering. Having the ADSM server doing the file filtering can potentially lead to reduced restore times.

2.2.8.3 Policy Definitions

To make point-in-time restores possible, the backup copy group VEREXIST and VERDELETED values should be set to nolimit to cover situations where backups are run more frequently than once daily. RETEXTRA and RETONLY should be set high enough to cover the desired point-in-time interval.

As an example, if the desired point-in-time interval is one month, the following backup copy group definitions are required:

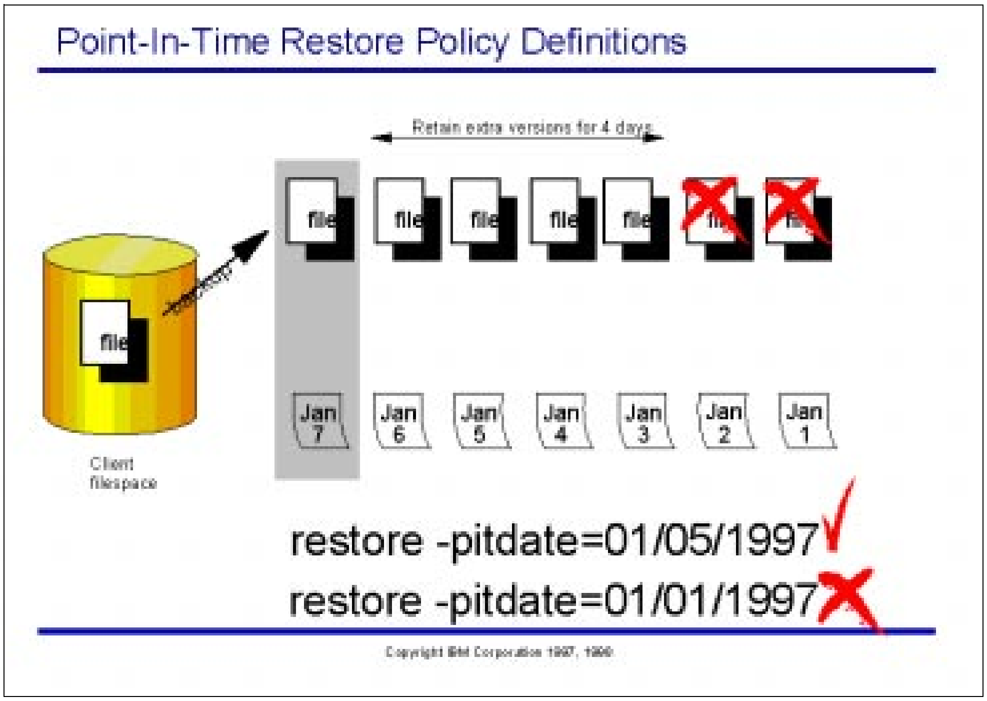
VEREXIST Nolimit

VERDELETED Nolimit

RETEXTRA 31 days or more

RETONLY 31 days or more

The impact that an increase in the retention period and the number of stored backup versions has on the size of the ADSM database should be taken into consideration before the policy definitions are changed. Copy group values with no limits such as VEREXIST and VERDELETED can result in large numbers of backup copies being maintained by the server. This is compounded in situations where users can run multiple incremental backups daily. If such values are to be used, use of the copygroup FREQUENCY parameter, to control the interval between incremental backups, should be considered.



Point-In-Time Restore Policy Definitions

restore -pitdate=01/05/1997✓

restore -pitdate=01/01/1997✗

Copyright IBM Corporation 1997, 1998

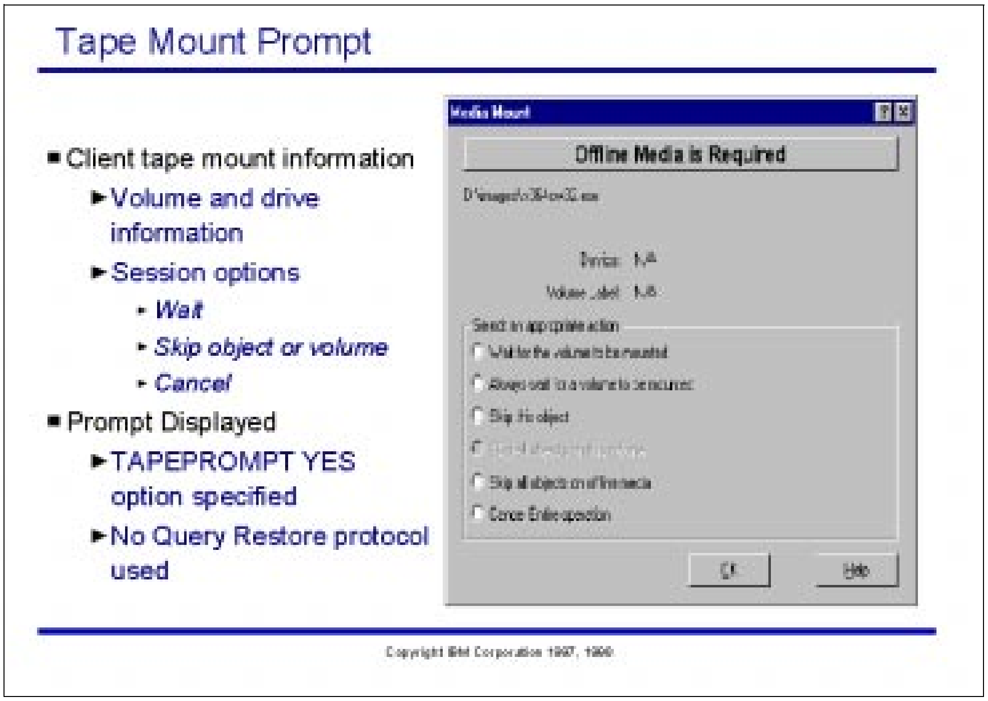
2.2.9 Point-in-Time Restore Policy Definitions

This graphic shows the effect of the ADSM backup copygroup definitions on the ability to perform a point-in-time restore. The following policy definitions are assumed:

- The VEREXIST and VERDELETED backup copygroup definitions are set to unlimited
- The RETEXTRA and RETONLY backup copygroup definitions are set to 4 days

Assuming that the file in the client filespace is backed up daily, a point-in-time restore is possible using 01/05/1997 as the point-in-time date.

A point-in-time restore is not possible using 01/01/1997 as the point-in-time date because the oldest backup version of the file is dated 01/03/1997. This is determined by the RETEXTRA and RETONLY definitions, in this example set to 4 days. Backup copies of the file more than 4 days old will have been expired on the ADSM server.



Tape Mount Prompt

- Client tape mount information
  - Volume and drive information
  - Session options
    - *Wait*
    - *Skip object or volume*
    - *Cancel*
- Prompt Displayed
  - TAPEPROMPT YES option specified
  - No Query Restore protocol used

Media Mount

Offline Media is Required

[illegible]

Device: N/A

Volume Label: N/A

Select an appropriate action

○ Wait for the volume to be mounted

○ Always wait for a volume to be mounted

○ Skip this object

○ [illegible]

○ Skip all objects on offline media

○ Cancel Entire operation

Copyright IBM Corporation 1997, 1998

2.2.10 Tape Mount Prompt

Currently, a user has limited control over how tape mounts required during a session are handled. The tape mount dialog is enhanced in Version 3, giving the user a much broader range of options to control how the mounting of sequential media is performed.

2.2.10.1 Client Tape Mount Information

The new tape mount prompt is available in the backup-archive client GUI and command line client. The prompt displayed by the command line client contains the same information as the GUI version:

```
--- Offline Media is Required ---
The following object requires offline media to be mounted.
      Object: D:\CLIENT\data.txt
      Device: N/A
Volume Label: N/A

Select an appropriate action
  1. Wait for the volume to be mounted
  2. Always wait for a volume to be mounted
  3. Skip this object
  4. Skip all objects on this volume (Not Supported)
  5. Skip all objects requiring a media to be mounted
  6. Abort entire operation
Action 1,2,3,4,5,6 :
```

The user is presented with server volume and drive information, and new options for the mount request:

Wait Wait for mount of the sequential volume.

Always wait Wait for mount of the sequential volume and all subsequent sequential volumes.

Skip object Skip the current file and continue processing the next file.

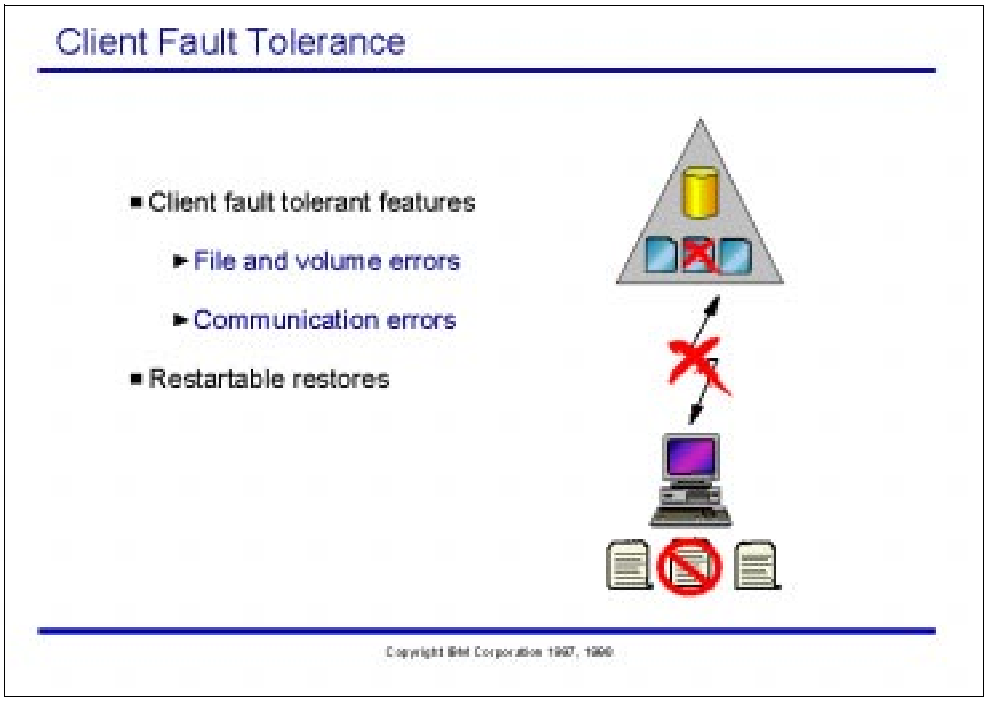
Skip tape Skip all files on the current sequential volume and continue processing files on the next sequential volume, if applicable.

Cancel Cancel the operation.

2.2.10.2 Prompt Displayed

The enhanced tape mount prompt is displayed when the TAPEPROMPT YES client option is used. The enhanced tape mount prompt is active for restore sessions only when the No Query Restore protocol is used.

2.3 Client Fault Tolerance



Client Fault Tolerance

- Client fault tolerant features
  - File and volume errors
  - Communication errors
- Restartable restores

Copyright IBM Corporation 1997, 1998

When a client operation fails in Version 2, a restart causes it to be started from the beginning. This is a performance problem at best, as processing already completed is repeated. Version 3 incorporates further fault-tolerant characteristics into clients, where the goal is to enhance the performance of the current fault tolerance and allow for a client operation's survival after a media or network failure.

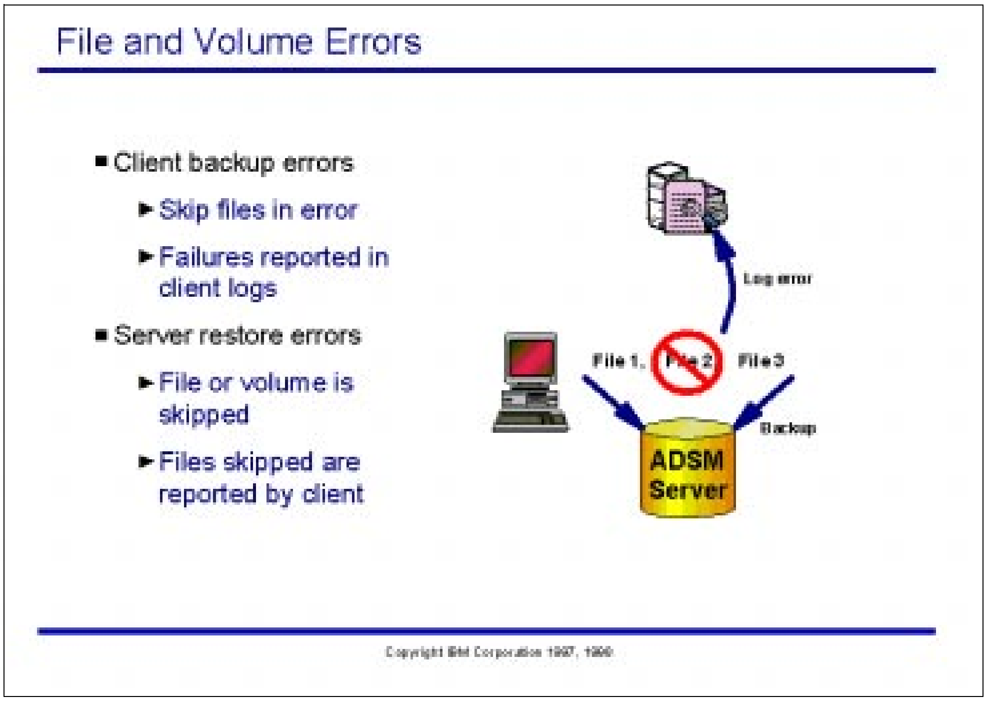
This section covers the following extensions to the fault-tolerant capabilities of the ADSM client:

- Client fault tolerant features

  The enhancements made to the backup-archive client provide support for errors such as communications errors, unreadable client files during backups, or volumes that are unavailable on the ADSM server during restore.

- Restartable restores

  Version 3 supports restore sessions that can be restarted from the point of failure following an interruption.



File and Volume Errors

- Client backup errors
  - Skip files in error
  - Failures reported in client logs
- Server restore errors
  - File or volume is skipped
  - Files skipped are reported by client

Log error
File 1,
File 2
File 3
Backup
ADSM
Server

2.3.1 File and Volume Errors

The Version 2 backup-archive client stops a backup or restore operation when file or volume errors are encountered. If the client operation is scheduled through the central scheduler in ADSM, the operation is restarted from the beginning as specified by the restart interval and the number of restart attempts specified by the schedule. The Version 3 backup-archive client is enhanced to handle file errors on the client and storage volume errors on the server by restarting the operation, beginning with the failing file.

2.3.1.1 Client Backup Errors

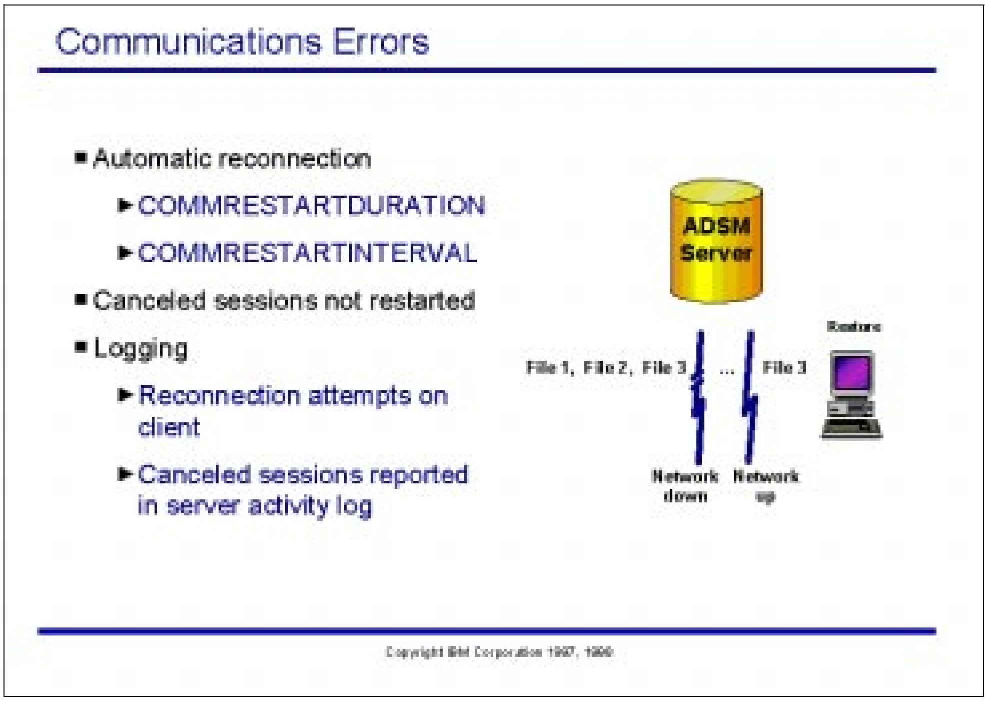
When a file-read error such as an unreadable file occurs, the Version 3 backup-archive client skips the file and continues processing the next file.

The failure is reported on the client in files such as the client error log and the schedule log. The error is optionally sent to the ADSM server through the central logging enhancements in Version 3.

2.3.1.2 Server Restore Errors

When a file is unavailable because of errors related to a storage pool volume during a restore operation, the file is skipped. If the volume where the file resides is unavailable on the ADSM server, all files on the volume are skipped.

The file or files skipped are reported back to the client where they are displayed on the screen and logged in the client error log and the schedule log.



Communications Errors

- Automatic reconnection
  - COMMRESTARTDURATION
  - COMMRESTARTINTERVAL
- Canceled sessions not restarted
- Logging
  - Reconnection attempts on client
  - Canceled sessions reported in server activity log

ADSM Server
Restore
File 1, File 2, File 3 ... File 3
Network down
Network up

Copyright IBM Corporation 1997, 1998

2.3.2 Communications Errors

The Version 2 backup-archive client had limited capabilities to recover from a communications error. If a client experiences a communications error, it has to be stopped and restarted to reestablish connection with the server. The Version 3 backup-archive client has new fault tolerant features that enable it to automatically reconnect to the server after a communications error.

2.3.2.1 Automatic Reconnection

If communication errors occur during a backup or restore operation, the client attempts to reopen the session. Two client options, COMMRESTARTDURATION and COMMRESTARTINTERVAL, determine how the communication restarts are handled. The options are in the client user options file (dsm.opt) on single-user clients and in the client system options file (dsm.sys) on multiuser clients.

COMMRESTARTDURATION This option determines the number of minutes during which the client tries to reconnect to the ADSM server after a communications error.

COMMRESTARTINTERVAL This option determines the number of seconds the ADSM client waits between reconnection attempts. The default is 15 seconds.

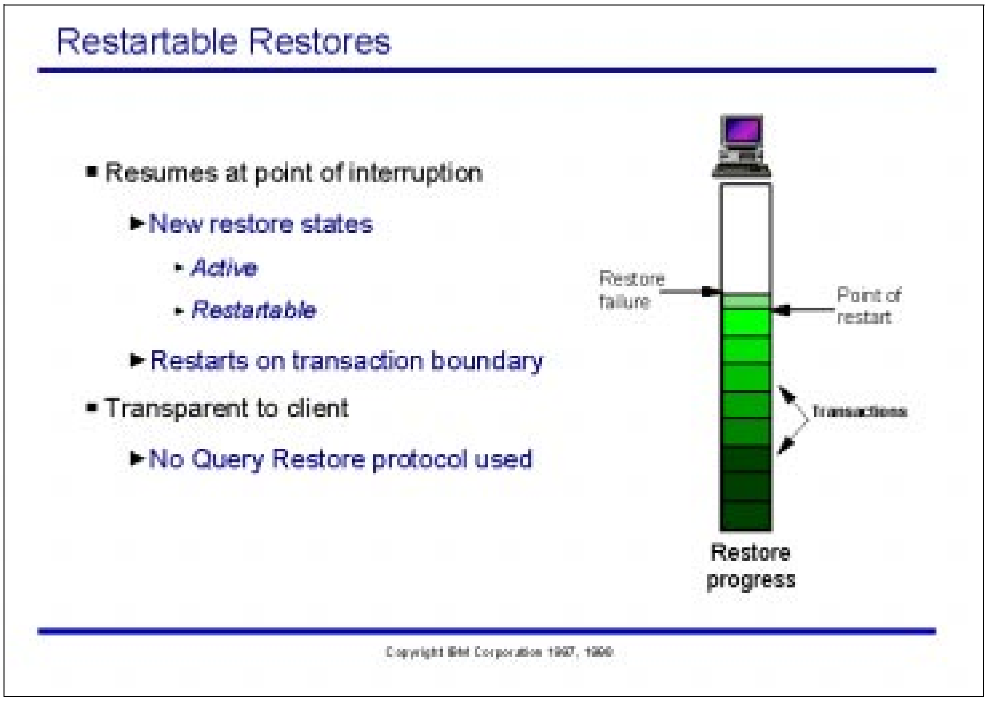
2.3.2.2 Canceled Sessions Not Restarted

If an administrator cancels a client session, the client receives a return code indicating that it was canceled on the first reopen attempt and exits without further automatic reconnection attempts.

2.3.2.3 Logging

The ADSM backup-archive client logs the reconnection attempts and displays them on the screen and in the client error log or the schedule log.

Information about canceled client sessions is stored in the activity log on the server.



Restartable Restores

- Resumes at point of interruption
  - New restore states
    - *Active*
    - *Restartable*
  - Restarts on transaction boundary
- Transparent to client
  - No Query Restore protocol used

Copyright IBM Corporation 1997, 1998

2.3.3 Restartable Restores

The fault-tolerant client features are designed to handle network and media-related errors more efficiently than in Version 2. The fault tolerance is enhanced even more with the concept of restartable restores (failed restore operations can be restarted from where they left off when the error situation occurred). The benefit of restartable restores is that processing completed before a failure does not have to be repeated.

2.3.3.1 Resumes at Point of Interruption

If an error occurs in the middle of a restore, the user can start another restore specifying the same source and destination. If the restore is started within the restart period allowed, the restore will start from where it left off. If a restore is restarted, some files may be restored again depending on how much of the ADSM transaction was complete when the error occurred.

Restartable restore sessions can have two states:

Active A restartable restore session that is running

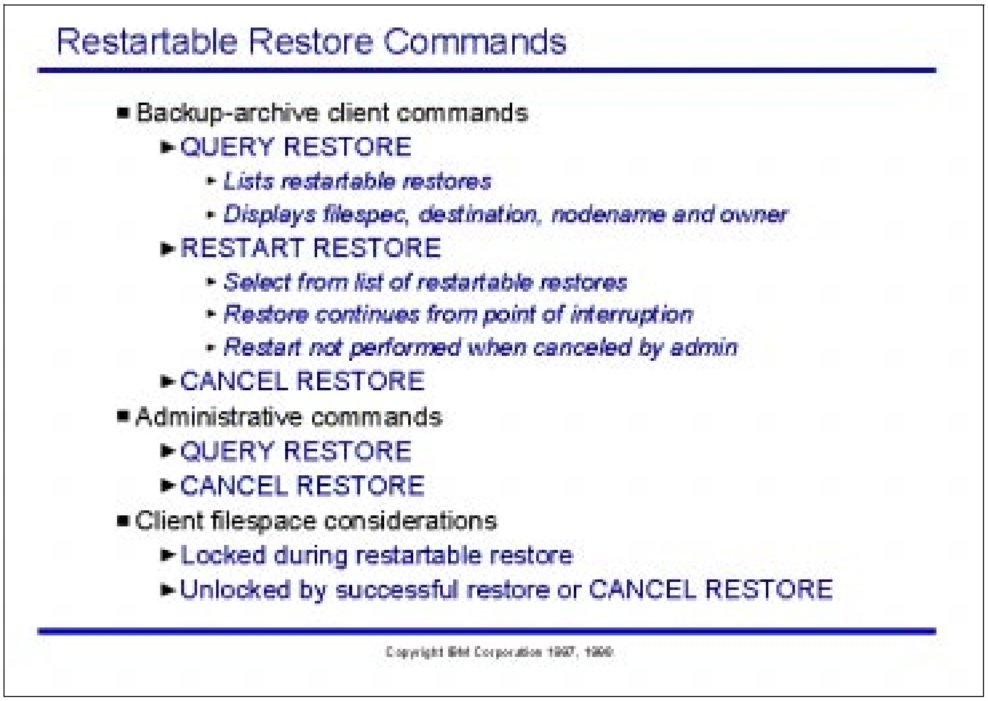
Restartable A restartable restore session that has been interrupted for some reason

Restore sessions are either in the active or restartable state. A restore session in progress is in the active state and is identified with the ADSM session number (from a QUERY SESSION/RESTORE). A session that has been interrupted is in the restartable state. The administrative QUERY RESTORE command shows all the restore sessions and their states. Restartable sessions displayed with this command have negative session numbers, making them easily distinguishable from active sessions.

A restarted restore starts at a transaction boundary as defined by the ADSM client and server options, TXNBYTELIMIT and TXNGROUPMAX.

2.3.3.2 Transparent to Client

The start of a restartable restore is transparent to a client. A restartable restore uses the No Query Restore protocol. A user cannot select a restartable restore (except to restart one). A message is displayed on the client node to indicate that a restartable restore session has started.



Restartable Restore Commands

- Backup-archive client commands
  - QUERY RESTORE
    - *Lists restartable restores*
    - *Displays filespec, destination, nodename and owner*
  - RESTART RESTORE
    - *Select from list of restartable restores*
    - *Restore continues from point of interruption*
    - *Restart not performed when canceled by admin*
  - CANCEL RESTORE
- Administrative commands
  - QUERY RESTORE
  - CANCEL RESTORE
- Client filespace considerations
  - Locked during restartable restore
  - Unlocked by successful restore or CANCEL RESTORE

Copyright IBM Corporation 1997, 1998

2.3.4 Restartable Restore Commands

Restartable restores introduce new backup-archive client commands, administrative commands, and considerations concerning client filespaces on the server.

2.3.4.1 Backup-Archive Client Commands

The backup-archive client GUI and command line client support restartable restores. The backup-archive client GUI contains a menu option to restart a restartable restore. New backup-archive client commands are provided to manage restartable restores.

- QUERY RESTORE

  Lists the active and restartable restore sessions:

```
dsmc> query restore
Node Name: TONGA
Session established with server YANGTZE: Windows NT
  Server Version 3, Release 1, Level 0.0
  Server date/time: 1997.07.19 05:54:36AM  Last access: 1997.07.19 05:54


List of restartable restore sessions:

    Owner        Start Date          REPLACE SUBDIR PRESERVEPATH
    ---------  -------------------   ------- ------ ------------
 1. *          1997.07.19 05:54:23AM Prompt  Yes    Subtree
    Source: C:\TEST\*     Destination:
```

- RESTART RESTORE

Restarts a restartable restore session. A user is presented with a list of restartable restore sessions that can be restarted:

```
dsmc> restart restore
Restore function invoked.


List of restartable restore sessions:

    Owner        Start Date          REPLACE SUBDIR PRESERVEPATH
    ---------  -------------------   ------- ------ ------------
 1. *          1997.07.19 05:54:23AM Prompt  Yes    Subtree
    Source: C:\TEST\*     Destination:

Which session to Restart ? (1 - 1 or Quit):
```

- CANCEL RESTORE

  Cancels a restartable restore session and removes it from the list of restore sessions. A user is presented with a list of restartable restore sessions that can be canceled:

```
dsmc> cancel restore

List of restartable restore sessions:

    Owner        Start Date          REPLACE SUBDIR PRESERVEPATH
    ---------  -------------------   ------- ------ ------------
 1. *          1997.07.19 05:54:23AM Prompt  Yes    Subtree
    Source: C:\TEST\*     Destination:

Which session to Cancel ? (1 - 1 or Quit):
```

Two identical restartable restores are not allowed at the same time. If a second restore operation is started specifying the same source and destination as the first restore, the following message is displayed:

```
07/14/1997 13:47:08 ANS1330S This node currently has a pending restartable
restore session.
The requested operation cannot complete until this session either
completes or is canceled.
```

2.3.4.2 Administrative Commands

Two new administrative commands are provided to support restartable restore sessions. These are administrative versions of the QUERY and CANCEL RESTORE commands.

The administrative QUERY RESTORE command reports active restores with the associated session number. Restartable restores are reported with negative session numbers, which makes them easier to identify:

```
adsm> query restore

Session    Restore        Elapsed    Node Name                        Filespace
           State          Minutes                                     Name
-------    -----------    -------    ------------------------         ---------
     -1    Restartable          0    TONGA                            OS2

adsm>
```

The administrator can use the CANCEL RESTORE command to cancel a restore in the restartable state:

```
adsm> cancel restore -1
```

This command removes the restartable restore session information from the server database and the lock on the client filespace.

> **Note**
>
> When the administrative CANCEL RESTORE command is used, the restore session number must be specified with the minus sign (-).

An administrator can also use the CANCEL SESSION command to cancel active restore sessions. This creates a restartable restore session that can be restarted by the backup-archive client:

CANCEL SESSION When an active restore session is canceled with the administrative CANCEL SESSION command, the backup-archive client can restart the restartable restore session, using the RESTART RESTORE command.

CANCEL RESTORE When a restore session is canceled with the administrative CANCEL RESTORE command, the backup-archive client cannot restart the session.

The server does not track the owner (on UNIX clients only) of client sessions that are canceled by an administrator. If more than one restartable restore session is active on a multiuser client, and an administrator cancels one of the restartable restore sessions, the first session that is subsequently started receives a return code from the server indicating that it was canceled by an administrator.

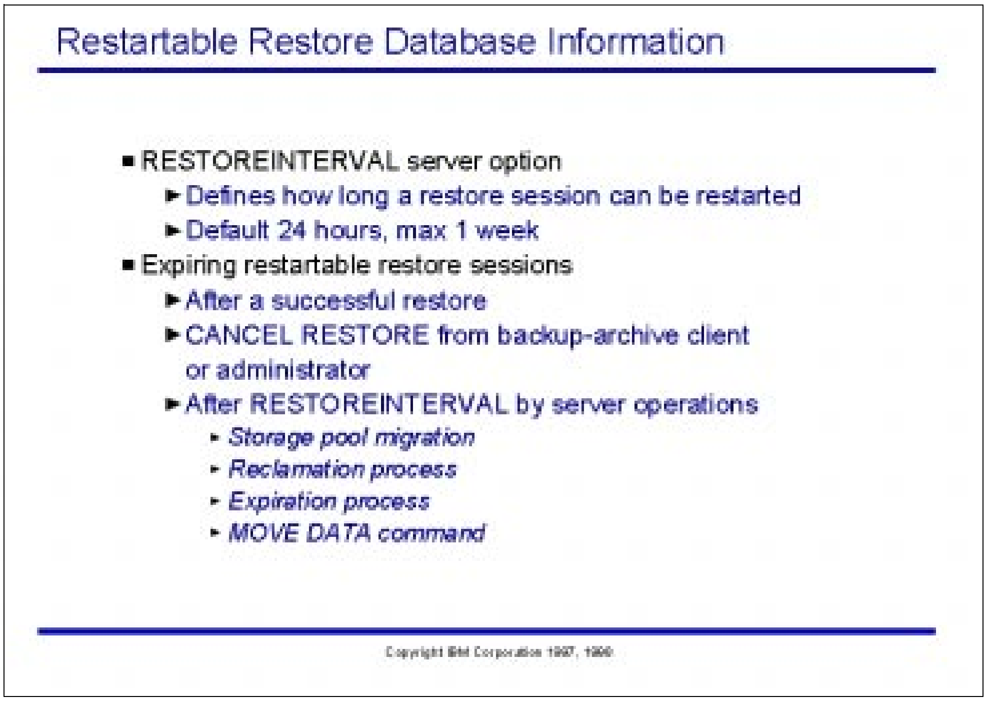
2.3.4.3 Client Filespace Considerations

To ensure a consistent restore, the client filespace that was being restored is locked while a restartable restore session is active. All server operations for sequential media volumes containing files for that filespace are prevented. The following server functions and administrative commands are affected:

- Storage pool migration
- Tape reclamation
- MOVE DATA commands

Further client backups for the locked client filespace are also prevented while a restartable restore session is active.

A successful restore, or a restore that is canceled by an administrator or backup-archive client, unlocks the client filespace.



Restartable Restore Database Information

- RESTOREINTERVAL server option
  - Defines how long a restore session can be restarted
  - Default 24 hours, max 1 week
- Expiring restartable restore sessions
  - After a successful restore
  - CANCEL RESTORE from backup-archive client or administrator
  - After RESTOREINTERVAL by server operations
    - *Storage pool migration*
    - *Reclamation process*
    - *Expiration process*
    - *MOVE DATA command*

2.3.5 Restartable Restore Database Information

The states of restartable restore sessions are kept in the ADSM database to allow the client to restart the operation at the point of failure.

2.3.5.1 RESTOREINTERVAL Server Option

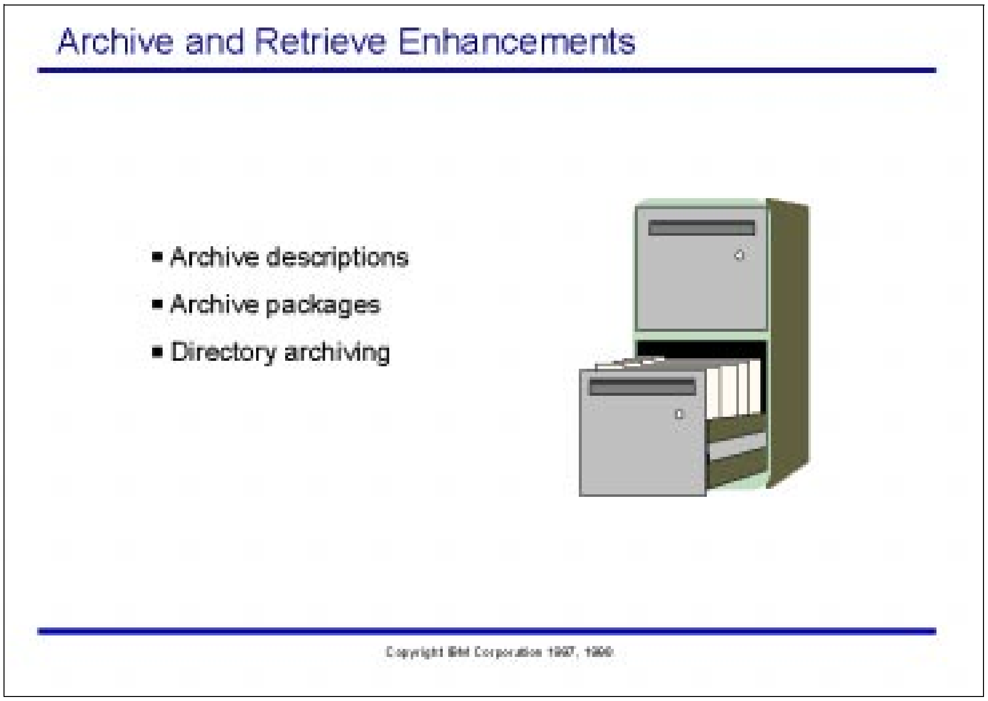
The new server option, RESTOREINTERVAL, sets the length of time restartable restore sessions are held in the restartable state. Information regarding the status of the restore is kept in the ADSM database. The default is to keep the restartable restore session information for 24 hours. RESTOREINTERVAL is defined in the server options file (dsmserv.opt) and can be changed dynamically.

2.3.5.2 Expiring Restartable Restore Sessions

The restartable restore state is removed from the ADSM database after a successful restore or when the restore is canceled by either the backup-archive client or an administrator.

The restartable restore state is also removed by some server processes after the RESTOREINTERVAL has elapsed. Server data movement operations such as storage pool migration, reclamation processes, expiration processing, and MOVE DATA commands remove the restartable restore state from the ADSM database when they run.

2.4 Archive and Retrieve Enhancements



Archive and Retrieve Enhancements

- Archive descriptions
- Archive packages
- Directory archiving

Copyright IBM Corporation 1997, 1998

This section covers the Version 3 enhancements to the archive function, with an emphasis on the new concept of archive packages. The following topics are covered:

- Archive descriptions

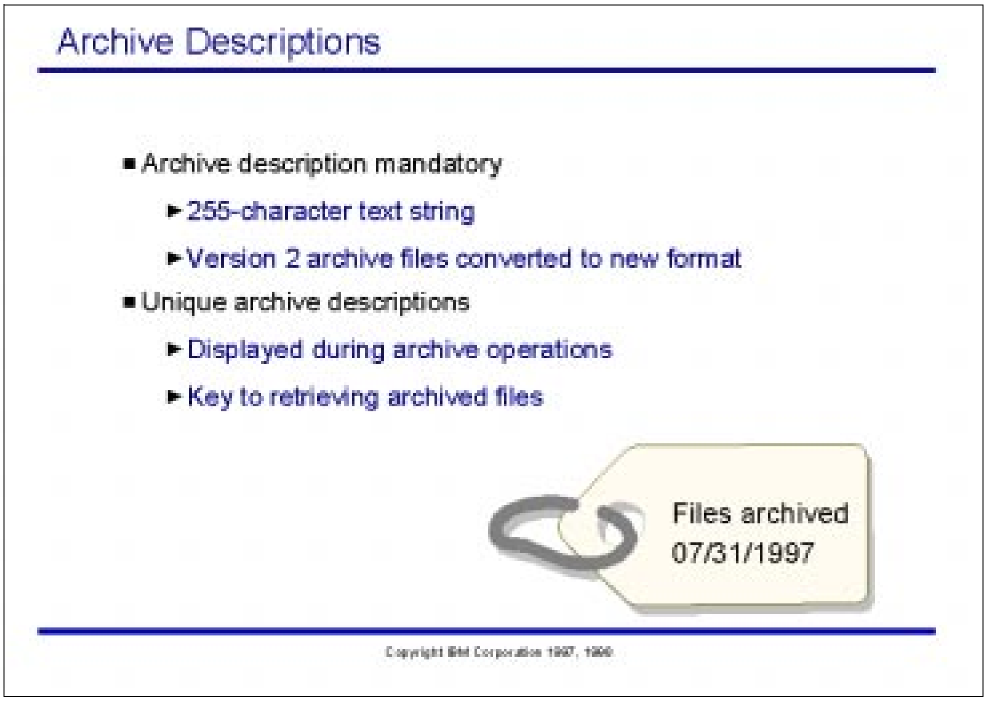
  An archive description is used to identify data through a meaningful description that can be utilized to identify the files at a later time. Version 3 requires that all archived files have an archive description.

- Archive packages

  A new way of grouping files and directories together is provided to make retrieval of archived files easier and more intuitive than in Version 2.

- Directory support

  Previous versions of ADSM did not support archive and retrieve of directories and their attributes. This support is now available in the Version 3 backup-archive client.



Archive Descriptions

- Archive description mandatory
  - 255-character text string
  - Version 2 archive files converted to new format
- Unique archive descriptions
  - Displayed during archive operations
  - Key to retrieving archived files

Files archived
07/31/1997

2.4.1 Archive Descriptions

In Version 2 the use of text descriptions for archive files was optional. With Version 3 archive descriptions are mandatory.

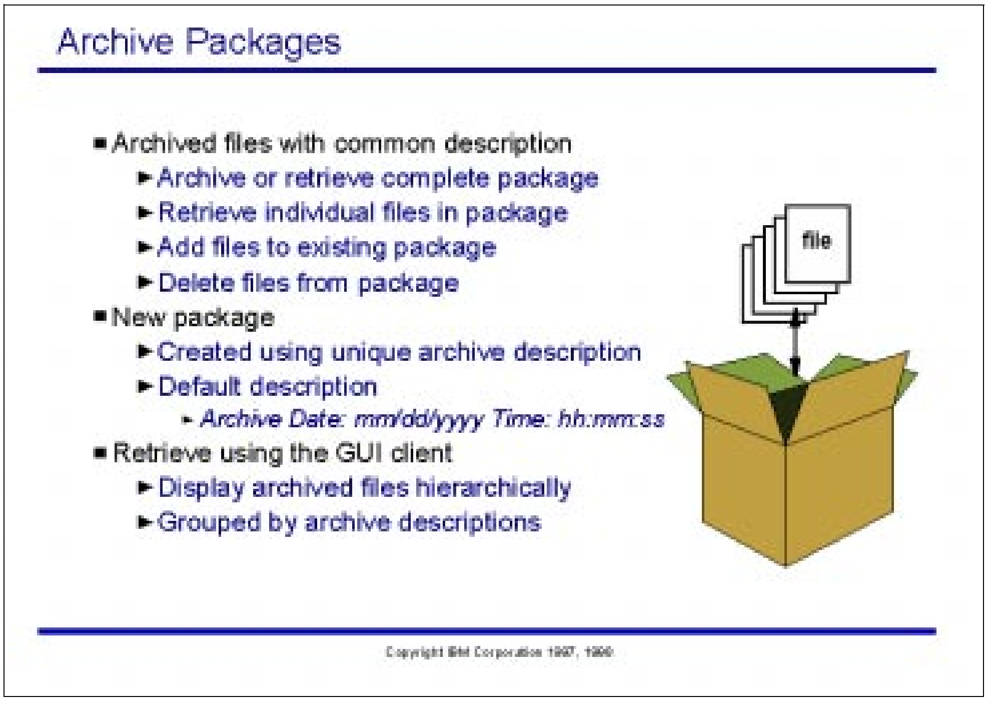
2.4.1.1 Archive Description Mandatory

When files are archived with the Version 3 backup-archive clients, an archive description is required. The archive description is a 255-character text string associated with files and directories. The archive description can be used to locate and identify files and directories without the knowledge of the physical client filespaces from which they were archived. If a description is not entered, a default archive description is assigned.

Version 3 stores the archive descriptions in a new format internally in the ADSM database. An initial conversion process occurs when a user starts an archive function. This conversion process can be long running if the user owns a large number of archived files. The conversion process can be canceled and restarted at a later time.

2.4.1.2 Unique Archive Descriptions

When the archive function is selected from the backup-archive GUI, a list of all previously used archive descriptions is displayed. The displayed archive descriptions can be used on subsequent archives.



Archive Packages

- Archived files with common description
  - Archive or retrieve complete package
  - Retrieve individual files in package
  - Add files to existing package
  - Delete files from package
- New package
  - Created using unique archive description
  - Default description
    - *Archive Date: mm/dd/yyyy Time: hh:mm:ss*
- Retrieve using the GUI client
  - Display archived files hierarchically
  - Grouped by archive descriptions

2.4.2 Archive Packages

Version 3 introduces archive packages. Archive packages are groups of files archived together with the same description.

2.4.2.1 Archived Files with Common Description

An archive package is a set of files and directories archived with a common, unique archive description. The association between archived files and their descriptions is much stronger in Version 3 than in previous versions. All files archived require a description in Version 3.

An archive package can be archived and retrieved as one entity. The Version 3 backup-archive client also supports:

- Retrieve individual files

  The files in a package can be retrieved individually.

- Add files in package

  Individual files can be added to an existing package by using the previously created package's archive description.

- Delete files from package

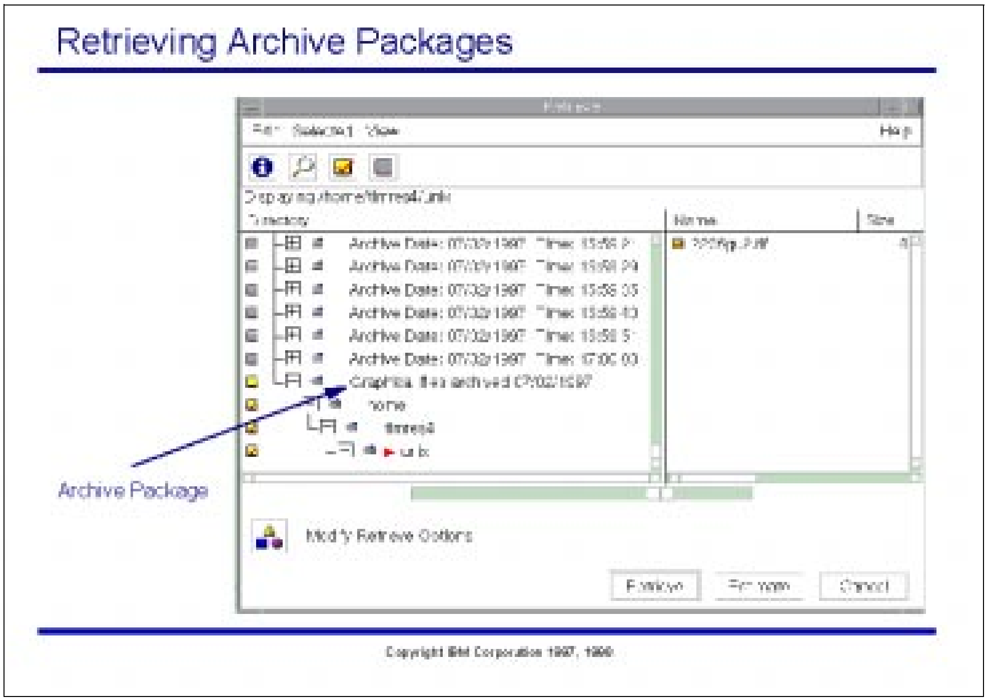
  Individual files can be deleted from a package.

2.4.2.2 New Package

A new package is created by specifying a unique archive description. The default description is: *Archive Date: mm/dd/yyyy Time: hh:mm:ss*. The date and time are NLS compliant.

2.4.2.3 Retrieve Using the GUI Client

The ADSM GUI retrieve function displays archived files hierarchically in a collapsible directory tree. The files are grouped by their archive descriptions. Expanding the collapsible description tree displays the individual directories and files of which the archive package consists. :hdref refid=2236g27. illustrates an example of retrieving archive packages with the GUI client.

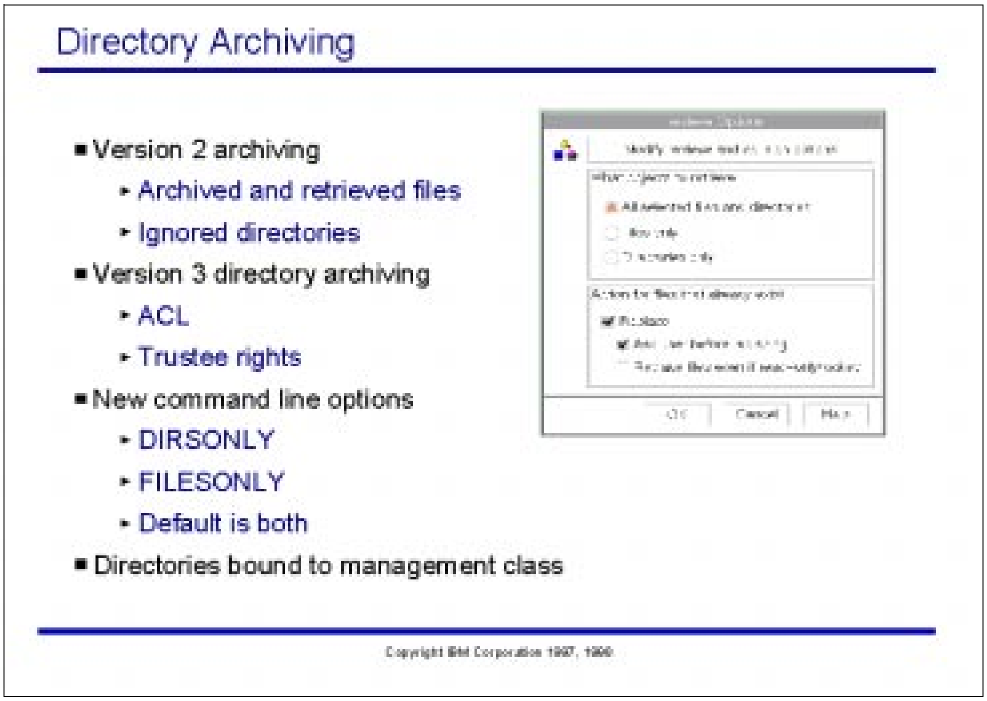


Retrieving Archive Packages

Archive Date: 07/02/1997 Time: 15:58:2
Archive Date: 07/02/1997 Time: 15:58:29
Archive Date: 07/02/1997 Time: 15:58:35
Archive Date: 07/02/1997 Time: 15:58:43
Archive Date: 07/02/1997 Time: 15:58:51
Archive Date: 07/02/1997 Time: 17:00:00
Graphics files archived 07/02/1997
home
timreg4
unix

2.4.3 Retrieving Archive Packages

The graphic shows the Version 3 Retrieve window where a list of packages is displayed (listed by archive description). One of the packages has been expanded and has had its file selected for retrieve.



Directory Archiving

- Version 2 archiving
  - Archived and retrieved files
  - Ignored directories
- Version 3 directory archiving
  - ACL
  - Trustee rights
- New command line options
  - DIRSONLY
  - FILESONLY
  - Default is both
- Directories bound to management class

2.4.4 Directory Archiving

The Version 3 archive function supports archive and retrieve of directories and their attributes.

2.4.4.1 Version 2 Archiving

The archive function in Version 2 does not support archive of directories. When files in a directory are archived, only the files themselves are archived. The directory structures and the attributes are ignored.

2.4.4.2 Version 3 Directory Archiving

The Version 3 backup-archive client supports archive and retrieve of directories. Directories with associated ACLs or trustee rights are archived when the files are archived and are retrieved when the files are retrieved.

Directories that are archived use the same archive description as the files with which they are archived.

2.4.4.3 New Command Line Options

New command line options are provided with Version 3 to support archive and retrieve of directories:

DIRSONLY When this option is specified, only the directories and their attributes are archived or retrieved.

FILESONLY When this command line option is specified, only the files and their attributes are archived or retrieved.

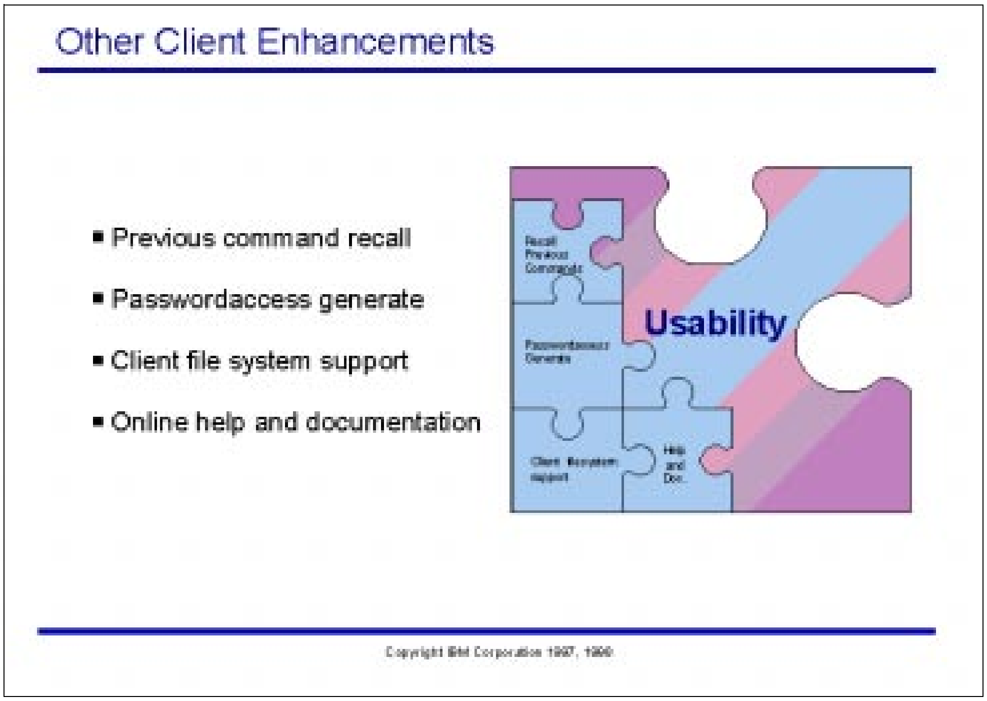
The default is to archive or retrieve both directories and files.

2.4.4.4 Directories Bound to Management Class

The process for determining the management class to use for the archived directories is the same as the process used for directories that are backed up:

1. The directories are bound to the management class referenced by the DIRMC backup-archive client option.
2. If the DIRMC option is not used, the management class with the longest retention period within the active policy set is used.

2.5 Other Client Enhancements



Other Client Enhancements

- Previous command recall
- Passwordaccess generate
- Client file system support
- Online help and documentation

Usability
Recall Previous Commands
Passwordaccess Generate
Client filesystem support
Help and Doc

This section discusses other miscellaneous Version 3 client-related enhancements:

- Previous command recall

  The Version 3 command line clients support recall and edit of previously entered commands.

- Passwordaccess generate support

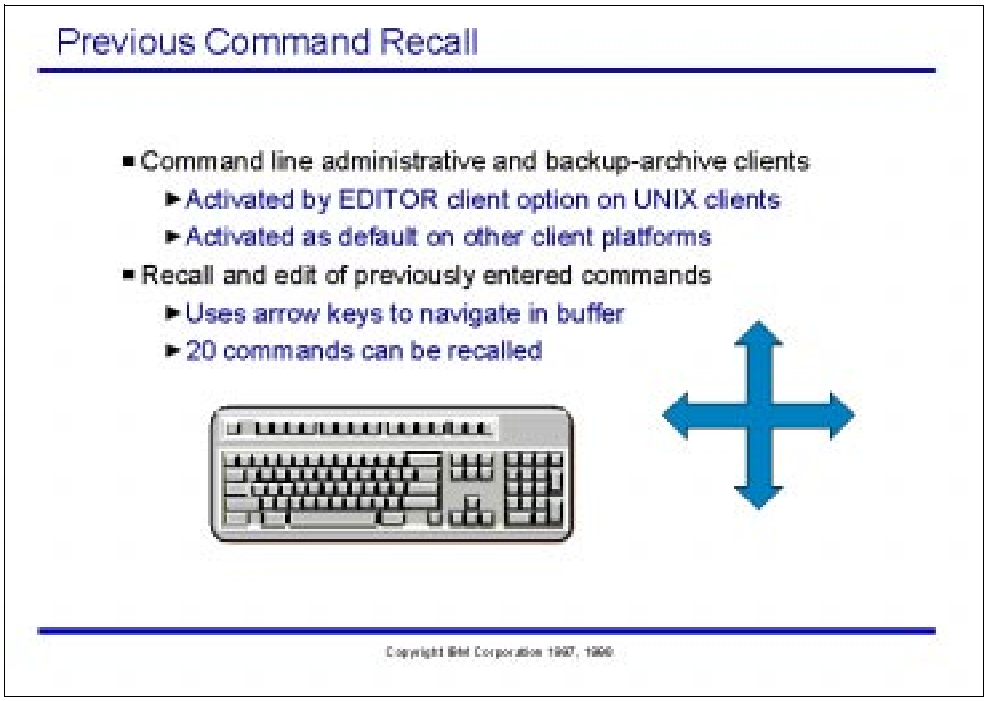
  Passwordaccess generate support simplifies management of the client/server authentication. This support came with Version 2 for certain client platforms but has been expanded to include other client platforms in Version 3.

- Client file system support

  The backup-archive client is enhanced to support backup and restore of files such as UNIX special files and Macintosh files on Windows NT servers.

- Online help and documentation

  The online help and documentation have been enhanced in Version 3. ADSM server and client messages have been translated to new languages, and support for cultural conventions such as date, time, and number formats has been added.



Previous Command Recall

- Command line administrative and backup-archive clients
  - Activated by EDITOR client option on UNIX clients
  - Activated as default on other client platforms
- Recall and edit of previously entered commands
  - Uses arrow keys to navigate in buffer
  - 20 commands can be recalled

2.5.1 Previous Command Recall

The Version 3 administrative and backup-archive command line clients have been enhanced to support recall and edit of previously entered commands.

2.5.1.1 Command Line Administrative and Backup-Archive Clients

The recall previous command support is available in the administrative and backup-archive clients on all client platforms. This support was previously provided by the client operating systems on certain platforms but is now provided by ADSM to expand to all client platforms.

On UNIX clients, support for recall of previous commands is activated by adding EDITOR YES to the client user options file (dsm.opt). On other client platforms, configuration options are not required to activate this support.

2.5.1.2 Recall and Edit of Previously Entered Commands

The keyboard is used to navigate in the command history buffer and edit the previously entered commands. The keyboard keys used have been standardized across all Version 3 command line client platforms. The up and down arrow keys are used to navigate in the command history buffer; for example, a user can select the previous or next command. The following keyboard keys are used:

F3 or Escape Exits the command line client

Ctrl C Aborts the command line client

Backspace Moves the cursor one position to the left and deletes the text (destructive delete)

Left arrow Moves the cursor one position to the left

Right arrow Moves the cursor one position to the right

Up arrow Displays the command last entered

Down arrow Displays the command entered after the command currently displayed

Tab Moves the cursor five positions to the right

Left Tab Moves the cursor five positions to the left

Ctrl left arrow or Ctrl L Moves the cursor to the beginning of previous word

Ctrl right arrow or Ctrl R Moves the cursor to the beginning of the next word

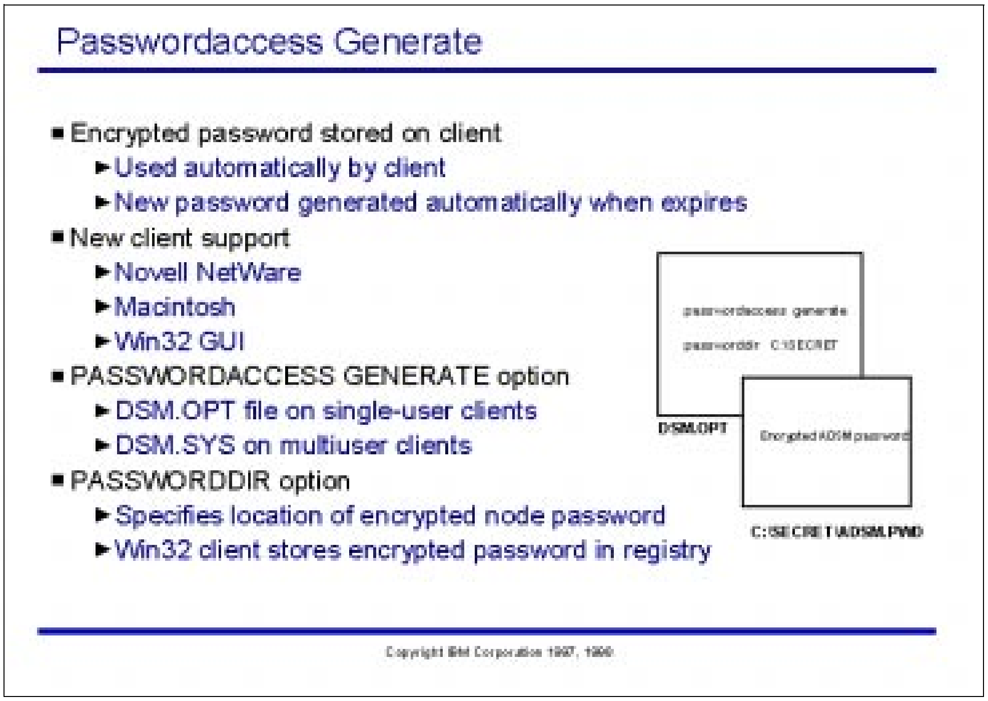
Home Moves the cursor to the beginning of the line

End Moves the cursor to the end of the line

Ctrl D or Ctrl Delete Deletes the text from the cursor to the end of line

Enter Executes the command

The Version 3 command line clients store up to 20 commands entered in a session in the command recall buffer. The recall previous command operates after a first-in-first-out (FIFO) principle; that is, the oldest command is removed when the buffer is full.



Passwordaccess Generate

- Encrypted password stored on client
  - Used automatically by client
  - New password generated automatically when expires
- New client support
  - Novell NetWare
  - Macintosh
  - Win32 GUI
- PASSWORDACCESS GENERATE option
  - DSM.OPT file on single-user clients
  - DSM.SYS on multiuser clients
- PASSWORDDIR option
  - Specifies location of encrypted node password
  - Win32 client stores encrypted password in registry

Copyright IBM Corporation 1997, 1998

2.5.2 Passwordaccess Generate

Client nodes are registered on the server with a password. The client uses this password when it authenticates with the server. The passwordaccess generate support automates the management of the client password and makes the authentication process between the client and server transparent.

2.5.2.1 Encrypted Password Stored on Client

The client's password is encrypted and stored in encrypted form in a file on the client. This stored password is used automatically by the client when it authenticates with the server. If a client receives a return code from the server indicating that the password has expired, it generates a random new password.

2.5.2.2 New Client Support

Version 3 supports passwordaccess generate on NetWare and Macintosh clients. Passwordaccess generate support has also been added to the Win32 backup-archive GUI. The Win32 backup-archive command line client and the Win32 scheduler previously had this support.

2.5.2.3 PASSWORDACCESS Generate Option

Passwordaccess generate support is activated by specifying PASSWORDACCESS GENERATE in a client options file. On single-user clients like OS/2, NetWare, and Windows NT, this option goes in the client user options file (dsm.opt). On multiuser operating systems like UNIX, it goes in the client system options file (dsm.sys).

2.5.2.4 PASSWORDDIR Option

The PASSWORDDIR backup-archive client option is used to specify the location of the encrypted password file. The Win32 backup-archive client is an exception to this rule as it stores the encrypted client password in the registry.

The backup-archive client uses the following rules to determine the location of the encrypted password file:

OS/2

1. The location specified by PASSWORDDIR
2. The DSM_DIR environment variable
3. The C:\ directory

UNIX

1. The location specified by PASSWORDDIR
2. If root protected:
   a. /etc/security/adsm for AIX else /etc/adsm
3. If non-root-protected:
   a. The DSM_DIR environment variable
   a. The directory from which the executable was loaded

NetWare

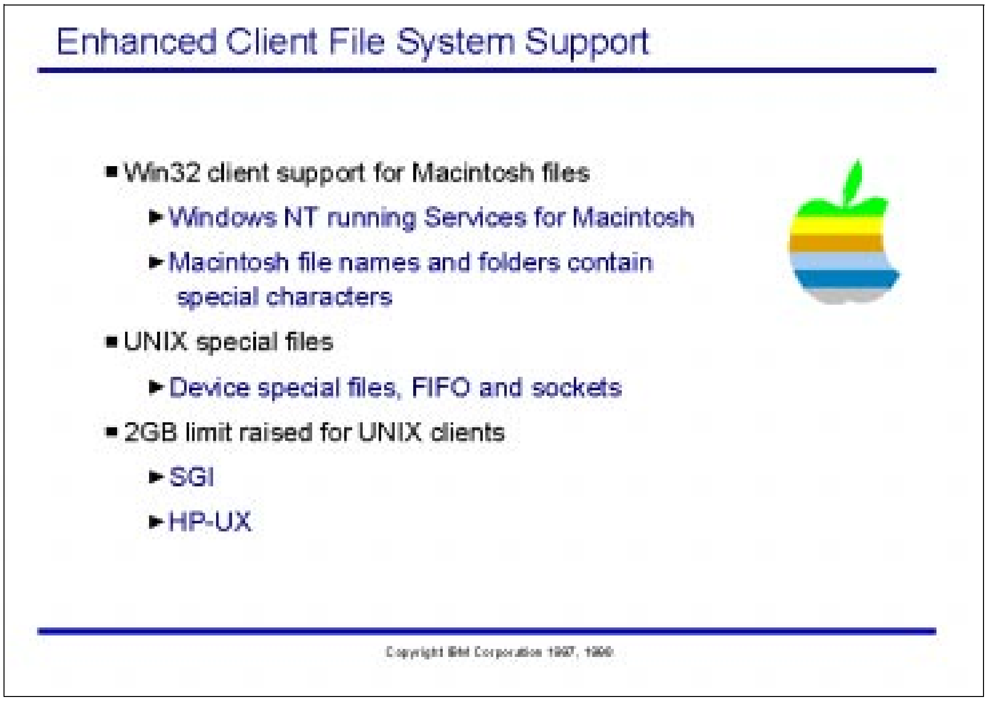
1. The location specified by PASSWORDDIR
2. The path from which DSMC.NLM is executing

Macintosh

1. The location specified by PASSWORDDIR
2. The ADSM folder within the Preferences folder on the system volume

Win32

1. The System Registry subtree



Enhanced Client File System Support

- Win32 client support for Macintosh files
  - Windows NT running Services for Macintosh
  - Macintosh file names and folders contain special characters
- UNIX special files
  - Device special files, FIFO and sockets
- 2GB limit raised for UNIX clients
  - SGI
  - HP-UX

Copyright IBM Corporation 1997, 1998

2.5.3 Enhanced Client File System Support

Version 3 introduces support for additional file systems and UNIX special files and increases file size support for UNIX clients.

2.5.3.1 Win32 Client Support for Macintosh Files

Customers running mixed Apple Macintosh and Windows NT environments and who use Windows NT as a file server must be able to back up and restore Macintosh files and folders located on the Windows NT servers.

Services for Macintosh software implement the Appleshare protocol on Windows NT servers and allow Windows NT to act as a file server for Apple Macintosh workstations. Macintosh files names and folders names can contain characters that are invalid in NTFS filespaces. Windows NT supports these invalid NTFS characters by changing them to Unicode. The Win32 backup-archive client now supports Macintosh files and folders by utilizing the Win32 Unicode API in Windows NT whenever backing up or restoring Macintosh files or folders.

2.5.3.2 UNIX Special Files

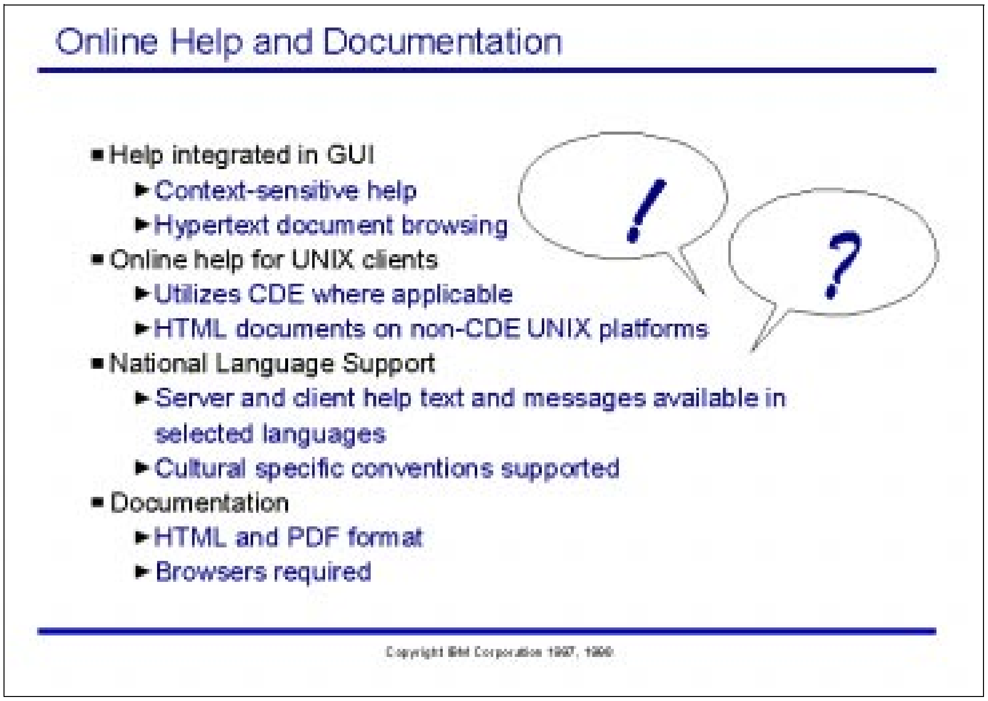
The Version 3 UNIX backup-archive clients are enhanced to support backup, restore, archive, and retrieve of UNIX special files, such as:

- Character special files
- Block special files
- FIFO special files
- Sockets

Hard links to device special files are supported.

2.5.3.3 2GB Limit Raised for UNIX Clients

Selected Version 2 backup-archive clients support backup and restore of files greater than 2GB. The support for files greater than 2GB has been extended to all UNIX clients where supported by the client operating system.



Online Help and Documentation

- Help integrated in GUI
  - Context-sensitive help
  - Hypertext document browsing
- Online help for UNIX clients
  - Utilizes CDE where applicable
  - HTML documents on non-CDE UNIX platforms
- National Language Support
  - Server and client help text and messages available in selected languages
  - Cultural specific conventions supported
- Documentation
  - HTML and PDF format
  - Browsers required

2.5.4 Enhanced Online Help and Documentation

A lot of emphasis has been put on making ADSM more user friendly. An important part of this process is the online help and documentation that is provided with the product.

2.5.4.1 Help Integrated in GUI

The online help system is integrated with the ADSM backup-archive client GUI to provide context-sensitive help and allow hypertext-based document browsing. The help system includes field level help and search facilities.

A Getting Started icon is provided to help first-time users with typical ADSM end user tasks. Video files in AVI/FLC format are used to emphasize usage of the backup-archive client on some client platforms.

2.5.4.2 Online Help for UNIX Clients

On UNIX clients that utilize the Common Desktop Environment (CDE), the backup-archive client uses the CDE help function to allow for functions like hypertext links and search functions. Currently these platforms are:

- AIX 4.x
- Sun Solaris 2.5
- HP-UX 10.x
- Digital UNIX

For other UNIX platforms, a help system based on Hypertext Markup Language (HTML) is provided. Use of this help system requires an HTML browser such as Netscape.

Non-UNIX clients use the standard help function provided by the operating system.

2.5.4.3 National Language Support

The national language support (NLS) in the server and clients allows ADSM to display help messages and text in languages other than English. Table 1 lists the ADSM componants and their supported languages.

Table 1. Version 3 Server and Client NLS Matrix

Platform	American English	Brazilain Portuguese	Italian	Chinese (Simplified)	Chinese (Traditional)	French	German	Japenese	Korean	Spanish	Swedish
AIX Server	X	X	X	X	X			X	X		
MVS Server	X							X			
NT Server	X			X	X			X			
AIX Client	X	X	X	X	X	X	X	X	X	X	X
HP-UX Client	X				X			X			
Solaris Client	X				X			X			
OS/2 Client	X	X	X	X	X	X	X	X	X	X	X
Win32 Client	X	X	X	X	X	X	X	X	X	X	X
Machintosh Client	X		X	X		X	X			X	X
NetWare Client	X		X	X		X	X			X	X
Other Clients	X										

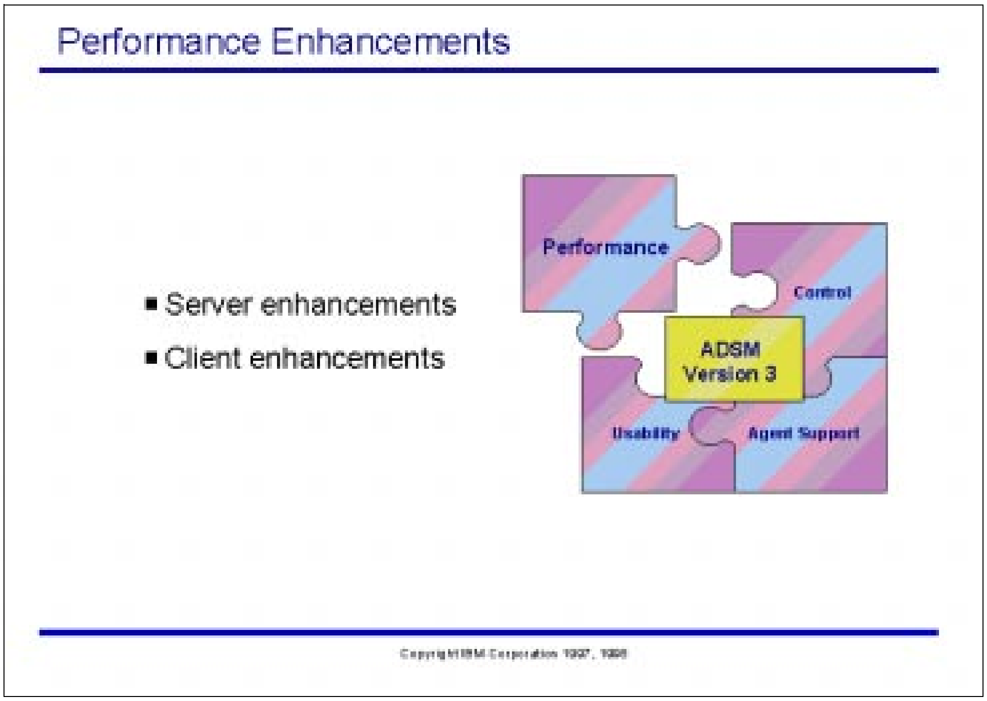
The Version 3 backup-archive client has been enhanced to use Unicode internally.

Cultural-specific conventions, also called *locales*, specify the date, time, and number formats to use. These conventions vary from country to country. For UNIX backup-archive clients, the locale support is provided through the XPG4 date and time functions.

2.5.4.4 Documentation

Online documentation is provided in HTML and Portable Document Format (PDF). To view the HTML documents, an HTML browser such as Netscape is required. A PDF browser such as the Adobe Acrobat Reader is required to view and print the PDF documentation. A free version of the Adobe Acrobat reader is provided with ADSM.

Chapter 3. Performance Enhancements



Performance Enhancements

- Server enhancements
- Client enhancements

Copyright IBM Corporation 1997, 1998

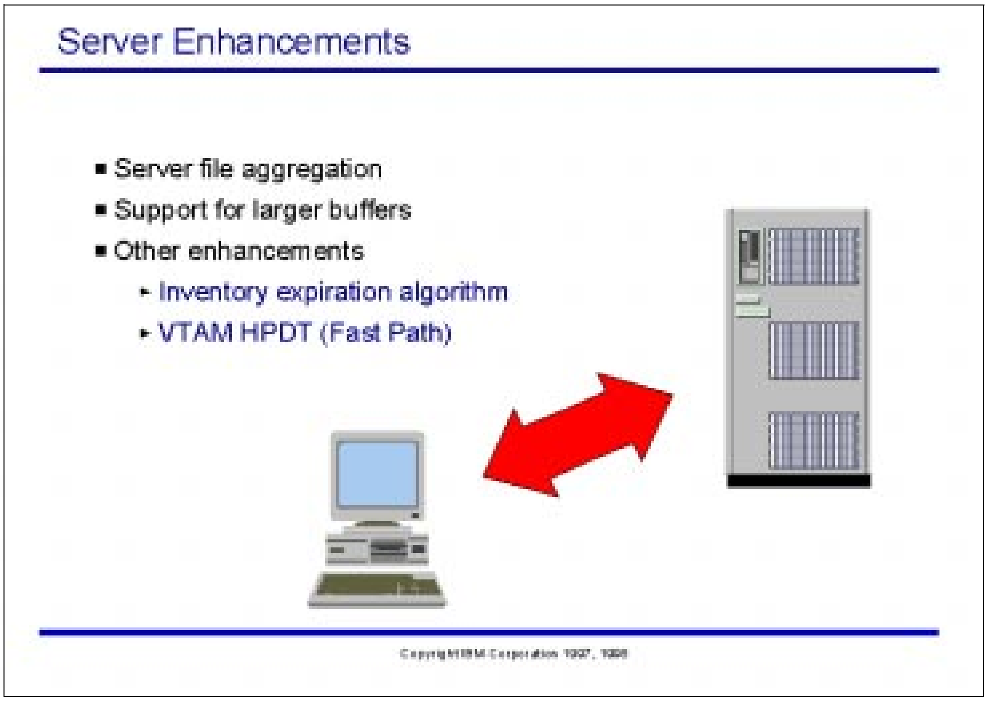
The exponential growth of distributed data has challenged network infrastructures and client/server applications to keep pace. As more and larger file servers and workstations proliferate, storage capacities grow at explosive rates. As devices and networks become faster and cheaper, the performance of client/server applications must meet the challenge.

To improve throughput, ADSM Version 3 implements new algorithms in small file aggregation and the inventory expiration process, improved buffering technologies, and a new restore process. Several new fault-tolerant features are provided such as the resumption of an interrupted restore without having to restart from the beginning.

This chapter covers the following topics:

- Server enhancements
- Client enhancements

3.1 Server Enhancements



Server Enhancements

- Server file aggregation
- Support for larger buffers
- Other enhancements
  - Inventory expiration algorithm
  - VTAM HPDT (Fast Path)

The following server enhancements are covered in this section:

- Server file aggregation

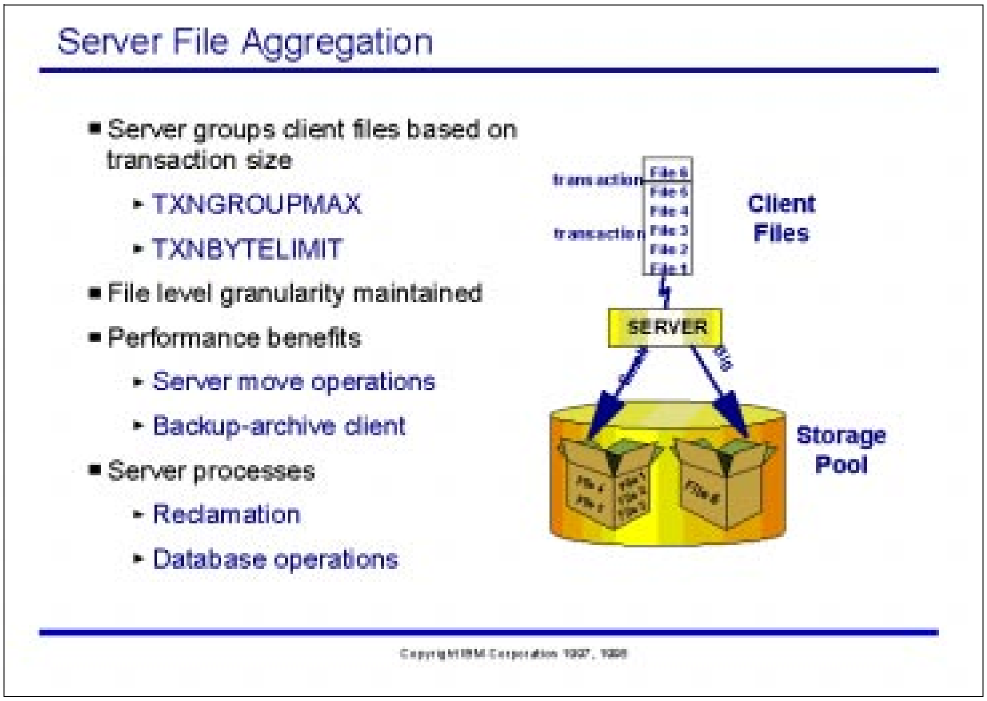
  Server file aggregation groups files together and significantly improves performance for certain operations, especially those operations where small files affect performance.

- Support for larger buffers

  The support for larger buffers is a component of the function commonly referred to as *fat pipes*. Support for larger buffers on the server is provided with all Version 3 servers. Tape blocking for the Version 3 servers increases the server device I/O pipe for larger capacity tape devices.

- Other enhancements

  The inventory expiration process has been improved, and support for the new VTAM 4.4 High Performance Data Transfer (HPDT or Fast Path) has been added for APPC on the MVS server.



Server File Aggregation

- Server groups client files based on transaction size
  - TXNGROUPMAX
  - TXNBYTELIMIT
- File level granularity maintained
- Performance benefits
  - Server move operations
  - Backup-archive client
- Server processes
  - Reclamation
  - Database operations

Copyright IBM Corporation 1997, 1998

3.1.1 Server File Aggregation

Server file aggregation is a new server function to group client files.

3.1.1.1 Server Groups Client Files Based on Transaction Size

The server can group multiple backup or archive files into an aggregate, which is a single physical file on the server. The size of an aggregate is based on the transaction from a client session. Two options, which have been available with previous versions of ADSM, determine the size of an ADSM transaction:

- TXNGROUPMAX is a server option that sets the maximum number of client files that a client can send to the server in a single transaction.
- TXNBYTELIMIT is a client option that sets the maximum number of bytes that a client can send to the server in a single transaction.

The first of these limits reached determines the size of the transaction and aggregate created on the server. Aggregates are comprised of complete files, so they are usually smaller than the TXNBYTELIMIT value specified. Client files that are larger than the TXNBYTELIMIT value are stored as a single physical file just as they are in earlier versions of ADSM. HSM migrated (space managed) files are not grouped into an aggregate.

In the example shown on the graphic, a client sends six files to backup or archive. As with previous versions, ADSM first determines the management class to which a file will be bound. Assume that all six files belong to the same management class. ADSM processes client files 1 through 5, known to ADSM as logical files, as a single transaction. The server aggregates these five files, as a single larger physical file, into the server storage pool. File 6 is larger than the TXNBYTELIMIT specified size and is stored as a single physical file on the server.

3.1.1.2 File Level Granularity Maintained

Other than the described TXNBYTELIMIT option, aggregation is transparent to the client from a usability perspective. File granularity within ADSM continues to be maintained. Query commands and server processes have been extended to support aggregates.

3.1.1.3 Performance Benefits

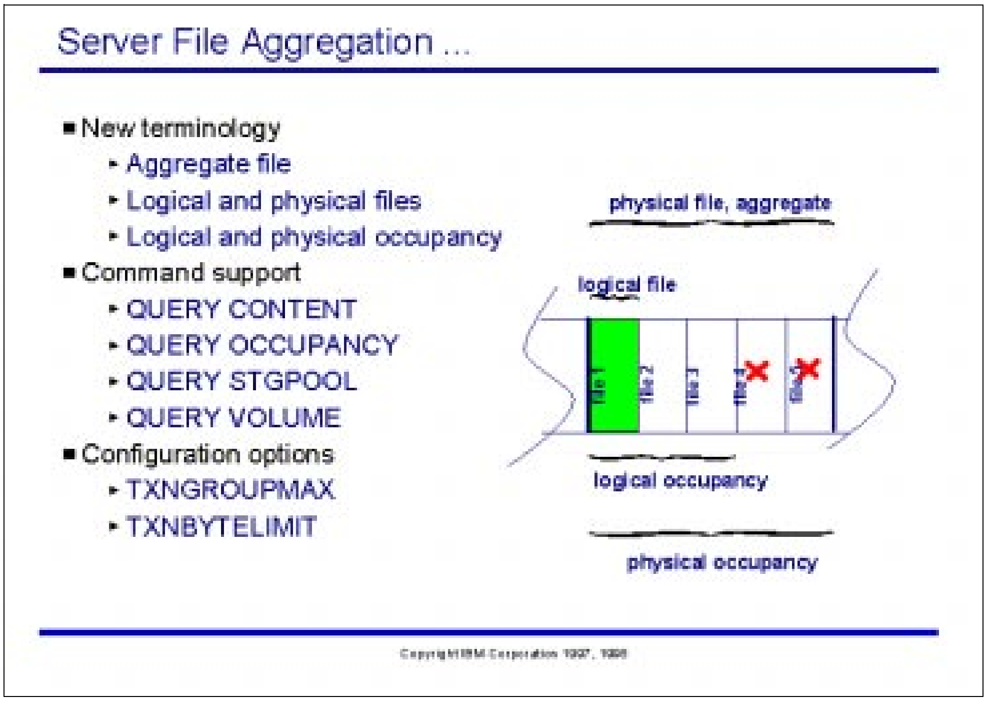
The performance of server processes that are a function of file size will be affected most by aggregation. Server processes that involve copy or move operations such as storage pool migration, storage pool backup and restore, and the move data operation are the primary beneficiaries of aggregation. The grouping of small client files into large physical files enables server processes to execute faster than previous versions of ADSM because of fewer database transactions needed to accomplish the same task.

Backup-archive sessions have also shown performance improvements as a result of aggregating small files. The greater the number of files that are aggregated, the fewer the number of database transactions required, compared to previous versions without aggregation. The database entries for a logical file within an aggregate are less than entries for a single physical file. Compared to previous versions of ADSM, this results in a side benefit of a smaller server database.

3.1.1.4 Server Processes

The reclamation process has been enhanced to handle aggregates. As logical files are deleted as the result of expiration, for example, unused space can occur within an aggregate. File granularity is maintained, and, because files within an aggregate may be deleted at different times, unused space within aggregates has to be handled. The RECLAIM parameter is still the control for reclamation processing. Logical occupancy is used to calculate the reclaimable space. Aggregates with unused space are compacted during the reclamation operation. The reclamation operation reclaims space on sequential access media only (random access storage pools are not reclaimed).

Because aggregates help to reduce the size of the database, certain operations involving the database run faster than in previous versions that do not aggregate files. For example, inventory expiration and database backup execute faster because the database is smaller.



Server File Aggregation ...

- New terminology
  - Aggregate file
  - Logical and physical files
  - Logical and physical occupancy
- Command support
  - QUERY CONTENT
  - QUERY OCCUPANCY
  - QUERY STGPOOL
  - QUERY VOLUME
- Configuration options
  - TXNGROUPMAX
  - TXNBYTELIMIT

3.1.2 Server File Aggregation ...

Server file aggregation introduces new terminology, commands, and configuration considerations.

3.1.2.1 New Terminology

Associated with server file aggregation is new terminology to describe client files and storage pool occupancy.

An aggregate file is stored in a storage pool and consists of a group of logical client files packaged together. The figure in the graphic shows five files that have been grouped together by the server to form an aggregate.

A logical file is a client file stored in a storage pool, either by itself, or as part of an aggregate file. Each of the five files in the figure is an individual and logical file from a client.

A physical file is stored in a storage pool and consists of either a single logical file or a group of logical files packaged together (an aggregate file). In the figure, logical files 1 through 5 comprise a single physical file (the aggregate file). If files 4 and 5 are deleted from within the aggregate, because they have been expired, for example, the physical file still includes the space occupied by files 4 and 5 until the reclamation process is run.

Logical occupancy is the space required for the storage of logical files in a storage pool. Because logical occupancy does not count the unused space created when logical files are deleted from aggregates, it may be less than physical occupancy. Once files 4 and 5 are deleted from the aggregate, the logical occupancy is reduced from the complete aggregate size to the space occupied by files 1 through 3.

Physical occupancy is the occupancy of physical files in a storage pool. It is the actual space required for the storage of physical files, including the unused space created when logical files are deleted from aggregates. In the figure, the physical occupancy of the aggregate is the space required to store files 1 through 5. Even after files 4 and 5 have been deleted, the physical occupancy remains the same until reclamation is run.

3.1.2.2 Command Support

The query commands are expanded to include information regarding aggregates:

- QUERY CONTENT FORMAT=DETAIL

  As with previous versions, the information returned lists the individual client files (logical files). Additional information returned includes whether the file is part of an aggregate and the number of logical files in the aggregate.

- QUERY OCCUPANCY

  Physical space occupied and logical space occupied have been added to the information returned because of aggregation. The logical space occupied information accounts for files within aggregates that have been expired or deleted. The reclamation process now also reclaims space within an aggregate.

- QUERY STGPOOL

  This command now also returns the logical occupancy of the utilized (not allocated) space to account for files within aggregates that have been expired or deleted. This number is a percentage of the utilized space, not allocated space. For example, consider a storage pool that has 100 MB allocated and 50 MB initially occupied with aggregates. The logical occupancy returned with the QUERY STGPOOL command is 100%. If half of the 50 MB becomes expired, the logical occupancy returned with this command is 50%.

- QUERY VOLUME FORMAT=DETAIL

  This command now displays the percentage of reclaimable space on a volume, which is calculated using the logical occupancy.

3.1.2.3 Configuration Options

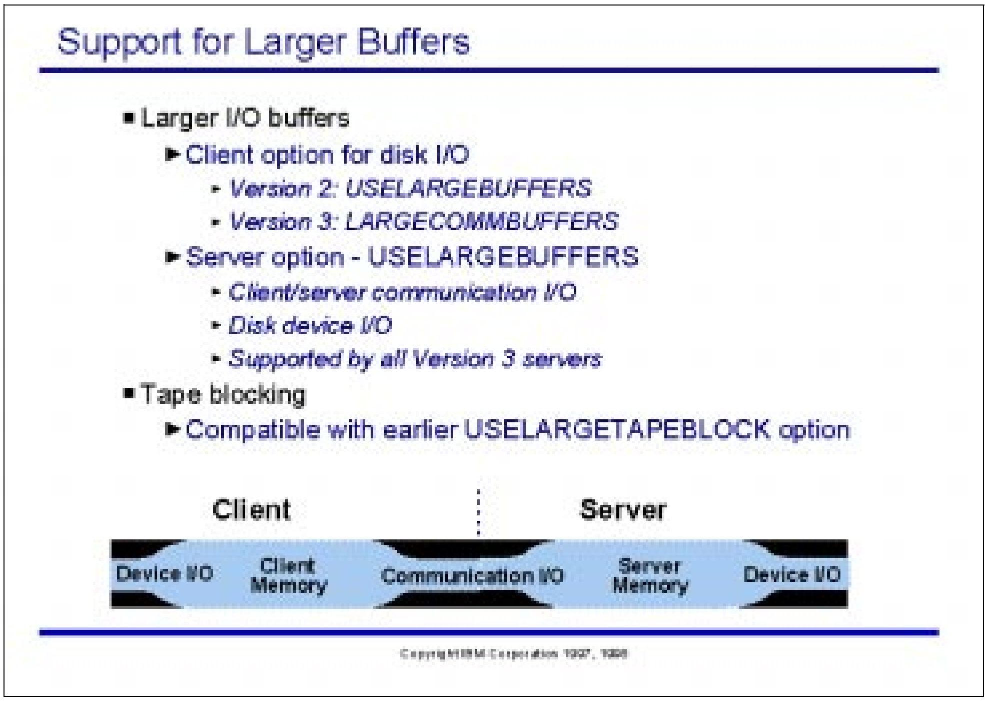
Server file aggregation is enabled by default in a Version 3 server. There are no external options to turn it off. The only configuration options for an administrator are the existing TXNGROUPMAX and TXNBYTELIMIT server and client options. These determine the size of the transaction between the server and client. The aggregate created is the same size as the transaction. Table 2 lists the default and maximum values for these options.

Table 2. TXNGROUPMAX and TXNBYTELIMIT Values

Option	Default	Maximum
TXNGROUPMAX	40 files	256 files
TXNBYTELIMIT	2048 KB	25600 KB

With these default settings the maximum aggregate is 40 files or 2048 KB of data, whichever condition is reached first. These default settings deliver some performance benefits, but increasing them further will deliver greater benefits. This performance improvement will particularly benefit file server clients with a large number of small files.

Many existing Version 2 users have increased the TXNGROUPMAX and TXNBYTELIMIT values to optimize backup-archive client network performance. Now that these options also determine the aggregate size, these options should be reviewed.



Support for Larger Buffers

- Larger I/O buffers
  - Client option for disk I/O
    - *Version 2: USELARGEBUFFERS*
    - *Version 3: LARGECOMMBUFFERS*
  - Server option - USELARGEBUFFERS
    - *Client/server communication I/O*
    - *Disk device I/O*
    - *Supported by all Version 3 servers*
- Tape blocking
  - Compatible with earlier USELARGETAPEBLOCK option

Client

Server

Device I/O

Client Memory

Communication I/O

Server Memory

Device I/O

Copyright IBM Corporation 1997, 1998

3.1.3 Support for Larger Buffers

Support for larger buffers is a component of the function commonly referred to as *fat pipes*. Support for larger buffers has been implemented at the communication I/O level on the server and separately at the device I/O level for the client and server.

3.1.3.1 Larger I/O Buffers

The USELARGEBUFFERS client option, available for AIX Version 2 clients in PTF 6, is used to increase the disk I/O buffers from 32 KB to 256 KB for a client reading or writing data from its disk. In Version 3 the client option has been renamed to LARGECOMMBUFFERS to avoid confusion with the server option, which is also called USELARGEBUFFERS. This feature is functionally the same as the Version 2 implementation.

The USELARGEBUFFERS server option, introduced for the AIX Version 2 server in PTF 12, is now included in all Version 3 servers and is the default setting. This option increases communication and device I/O buffers. Both the client/server communication buffer and disk device I/O buffers have been increased from 32 KB to 256 KB. The communication I/O buffer is used during data transfer with a client session, such as a backup session. The disk I/O buffer is used when data is read from or written to a disk storage pool.

Significant improvement in data transfer operations and CPU usage has been observed when the USELARGEBUFFERS feature is enabled. Increasing the buffer sizes allows client/server communications and disk I/O to be more efficiently performed than when this feature is not enabled. Reads and writes are quicker and server resources are better utilized. By reducing CPU utilization, more clients can concurrently be serviced, improving overall system performance. With Version 3 these benefits are complemented by aggregation.

Aggregation groups smaller, logical client files into fewer but larger physical files at the server level. Larger files are better able to take advantage of the larger buffers.

The USELARGEBUFFERS option, on by default, results in the server storing data in a new format. Files backed up on a server through this option are not accessible from a Version 2 server without the PTF that enabled this function. Data written in the old format can be read with this option enabled.

3.1.3.2 Tape Blocking

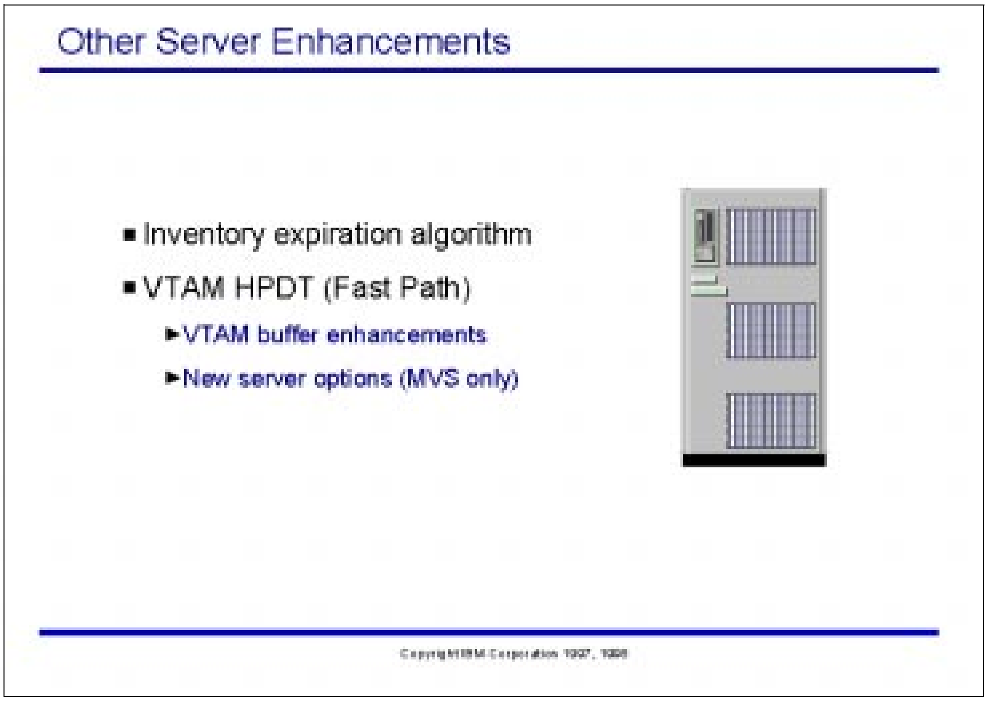
The USELARGETAPEBLOCK server option was first introduced in a PTF for the AIX and Sun Solaris Version 2 servers. This option increased the maximum tape block size for DLT devices from 32 KB to 256 KB. By writing larger blocks to the tape drive, tape operations are faster. Similar to the enablement of the USELARGEBUFFERS server option, data written to tapes with the USELARGETAPEBLOCK feature cannot be read by a server that does not support the feature. This might occur in Version 2 if the PTF is removed. Data that is exported is not affected, and data written in the previous format can be still be read with this option enabled. The option was added in Version 2 to allow administrators to make the decision to turn on the feature, because there are implications involved in using the feature.

In Version 3 this option is no longer used. There are no options to define the maximum tape block size. Table 3 details the maximum tape block sizes in Version 3 for the various device types.

Table 3. Version 3 Maximum Tape Block Sizes

Device Type	Maximum Block Size (KB)
QIC	32
4 mm	32
8 mm	32
3480/90	32
DLT	256
DTF	256
3570	256
3575	256

As with the USELARGETAPEBLOCK option in Version 2, backward compatibility is maintained. A Version 3 server can read tape created on Version 2 with smaller block sizes. However, Version 2 cannot read tapes created on a Version 3 server with larger block sizes.



Other Server Enhancements

- Inventory expiration algorithm
- VTAM HPDT (Fast Path)
  - VTAM buffer enhancements
  - New server options (MVS only)

3.1.4 Other Server Enhancements

3.1.4.1 Inventory Expiration Algorithm

The inventory expiration process has a new optimized algorithm. In addition to other ADSM database code improvements, this new algorithm is designed to provide improvements in the expiration process.

3.1.4.2 VTAM HPDT (Fast Path)

Support for the new VTAM 4.4 High Performance Data Transfer (HPDT or Fast Path) is added for the MVS ADSM servers that use the APPC communications method. The HPDT service optimizes the performance of large message transfers for APPC applications, which ADSM can exploit when using the APPC protocol. This service reduces the number of times data is moved between VTAM and an application (ADSM) when servicing a send or receive request. VTAM HPDT uses its Common Storage Manager (CSM) to hold buffers for application data, which can be accessed directly by applications such as ADSM. CSM uses MVS virtual storage in either data spaces or in the Extended Common Service Area (ECSA).

Support for VTAM HPDT introduces new MVS server options:

HPDTENABLE Used to enable or disable HPDT services

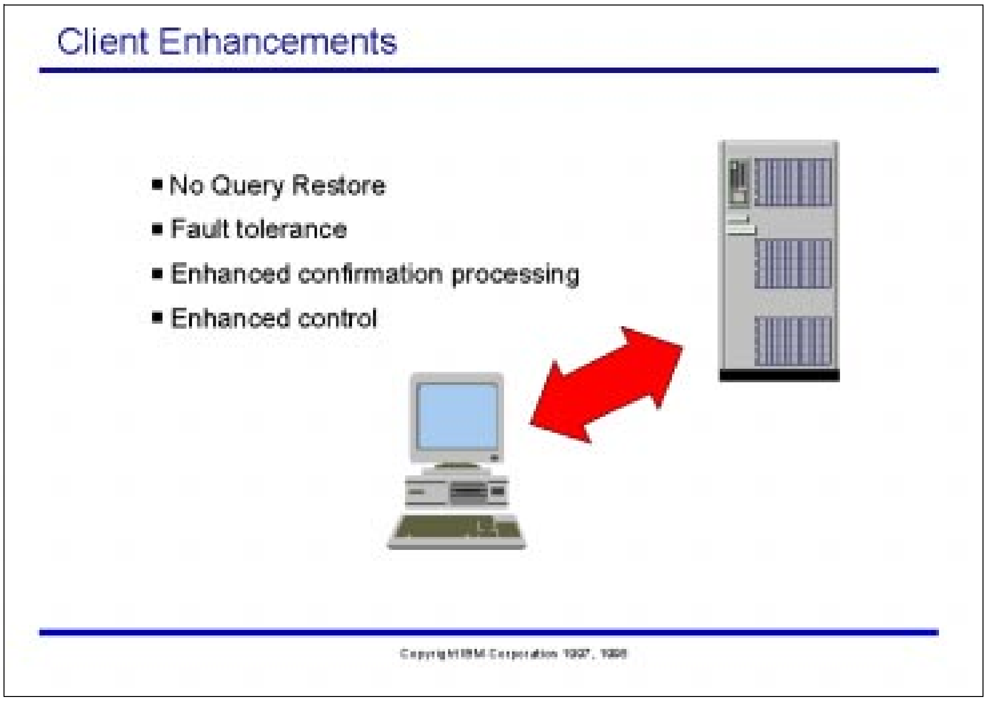
HPDTSTORAGE Defines what type of CSM buffers to use (data spaces or ECSA)

HPDTBUFFER# Specifies the number of CSM buffers to allocate initially

HPDTEXPAND# Specifies the number of additional CSM buffers to allocate once the initial allocation is used and additional buffers are required

If you use this new facility in VTAM, the overhead per ADSM transaction should be reduced, which would improve backup performance for a client.

3.2 Client Enhancements



Client Enhancements

- No Query Restore
- Fault tolerance
- Enhanced confirmation processing
- Enhanced control

Copyright IBM Corporation 1997, 1998

This section covers the following client-related enhancements:

- No Query Restore

  No Query Restore is a new restore protocol that reduces client memory requirements and enables restarting failed restore sessions.

- Fault tolerance

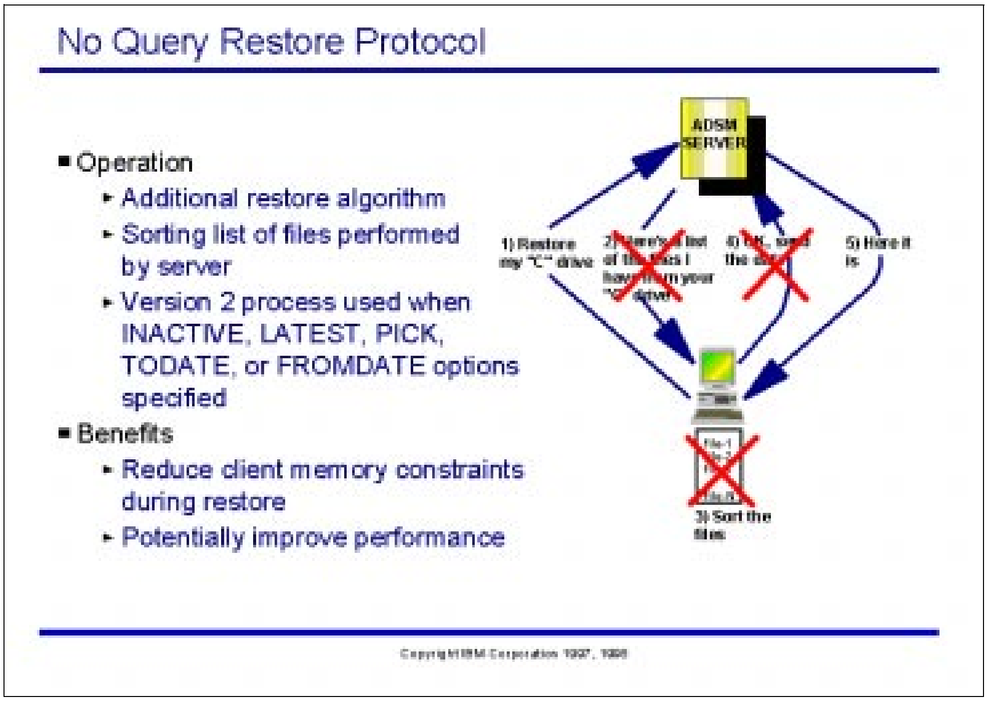
  In addition to the restartable restore, clients are more tolerant of communication interruptions and unreadable files that may occur from file or volume errors.

- Enhanced confirmation processing

  A confirm is an ADSM client-generated handshaking protocol used during client/server sessions. This process has been enhanced to generate more intelligent confirms, resulting in less overhead.

- Enhanced control

  Several usability enhancements allow for better granularity and navigation of directory structures, resulting in better control of data transfer, possibly reducing the amount of data to be moved and resulting in quicker overall execution.



No Query Restore Protocol

- Operation
  - Additional restore algorithm
  - Sorting list of files performed by server
  - Version 2 process used when INACTIVE, LATEST, PICK, TODATE, or FROMDATE options specified
- Benefits
  - Reduce client memory constraints during restore
  - Potentially improve performance

Copyright IBM Corporation 1997, 1998

3.2.1 No Query Restore Protocol

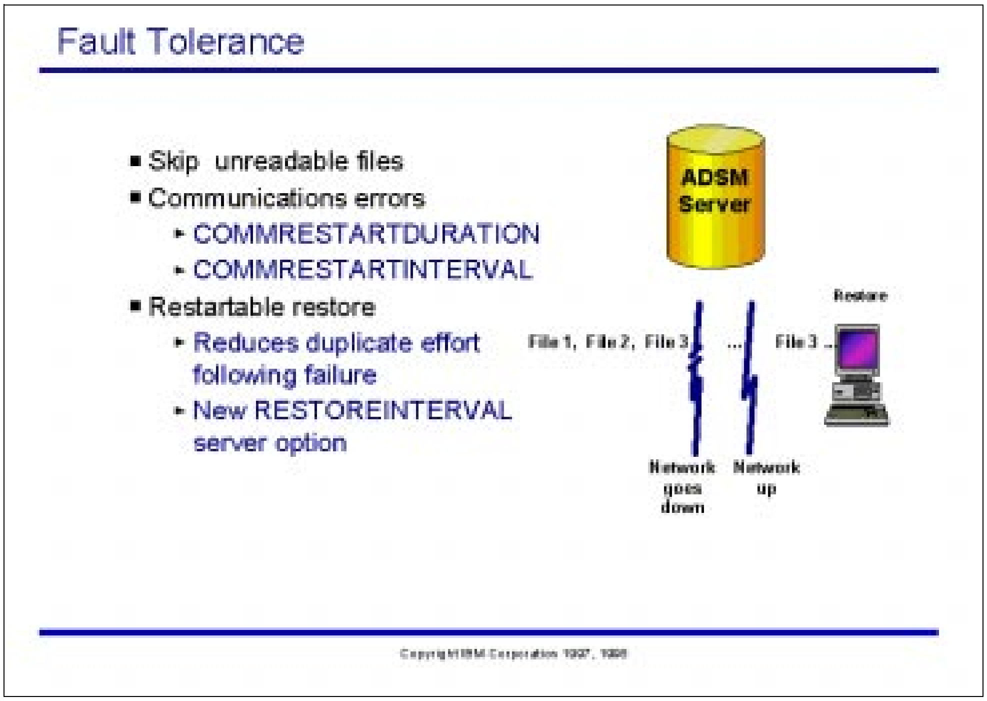
The No Query Restore protocol has already been described (2.2.4, “No Query Restore Protocol” on page 52). Here it is reviewed and the performance benefits discussed.

3.2.1.1 Operation

From a performance perspective, the difference in this restore process is that the list of files is no longer sorted on the client. With the No Query Restore protocol, the list of files to be restored is held, and controlled, by the server. There are no options to turn on to enable the No Query Restore protocol. This protocol is used for all restore operations unless one of the following options is used: INACTIVE, LATEST, PICK, TODATE, or FROMDATE. Use of any of these options invokes the Version 2 restore protocol.

3.2.1.2 Benefits

With the No Query Restore protocol client memory requirements are reduced. This reduction allows restores of large amounts of files that had previously failed because the creation of the list of files on the client required more memory than was available. Clients such as NetWare that have memory resource constraints, many directories, or many files to restore will receive the most benefit. Removing the file list sorting from the client also could potentially improve overall performance for the restore operation.



Fault Tolerance

- Skip unreadable files
- Communications errors
  - COMMRESTARTDURATION
  - COMMRESTARTINTERVAL
- Restartable restore
  - Reduces duplicate effort following failure
  - New RESTOREINTERVAL server option

Copyright IBM Corporation 1997, 1998

3.2.2 Fault Tolerance

While these items are not specifically designed to improve raw performance throughput, they do assist in improving the overall completion of client tasks, such as the backup of a client’s filespace. These client fault tolerance items help to reduce overall task completion time in situations where error conditions or interruptions require detection, correction, intervention, and repetition of data movement.

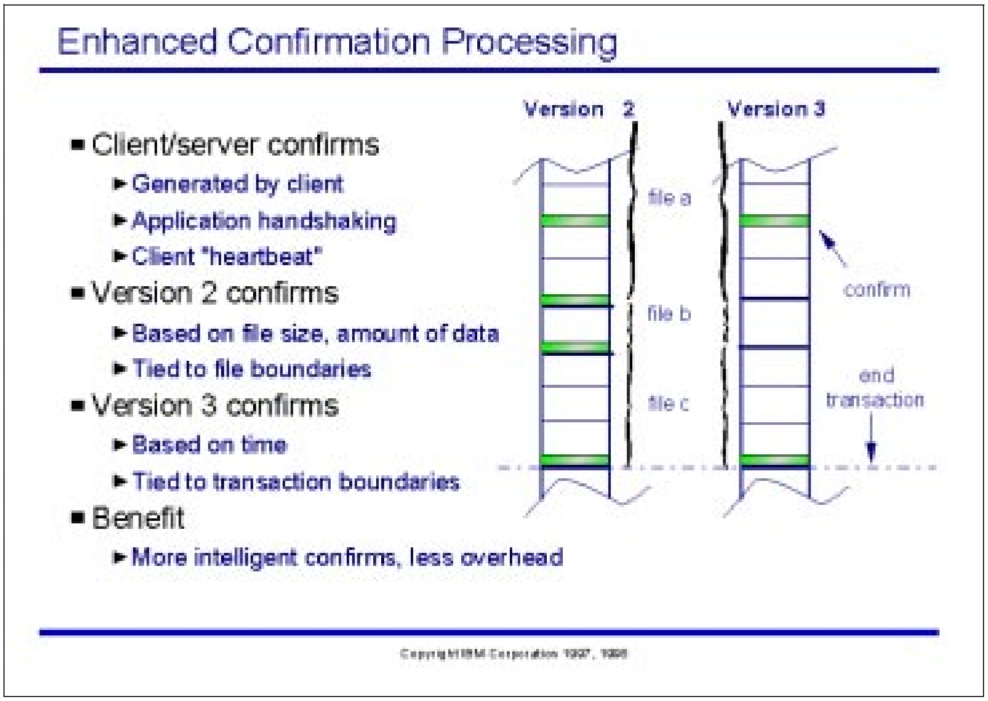
3.2.2.1 Skip Unreadable Files

Errors encountered during backup or restore that are due to a file or volume error problem are logged and then skipped. Thus the backup or restore process can continue and finish its task without requiring intervention and a restart.

3.2.2.2 Communication Errors

Version 3 clients are more tolerant of network connectivity interruptions. Two new options, COMMRESTARTDURATION and COMMRESTARTINTERVAL, control the restart window of time and interval between restarts. This enhancement assists in environments that are subject to heavy network congestion or frequent interruptions and eases the manageability of large numbers of clients by reducing intervention on error conditions.

Restartable restores allow restores to continue after an interruption without starting at the beginning. This reduces duplicate effort or manual determination of where a restore process was terminated. A new server option, RESTOREINTERVAL, defines the amount of time an interrupted restore can remain in the restartable state.



Enhanced Confirmation Processing

- Client/server confirms
  - Generated by client
  - Application handshaking
  - Client "heartbeat"
- Version 2 confirms
  - Based on file size, amount of data
  - Tied to file boundaries
- Version 3 confirms
  - Based on time
  - Tied to transaction boundaries
- Benefit
  - More intelligent confirms, less overhead

Copyright IBM Corporation 1997, 1998

3.2.3 Enhanced Confirmation Processing

Version 3 introduces changes to the method in which clients and servers communicate.

3.2.3.1 Client/Server Confirms

While a session is established between a client and a server, they exchange a succession of "handshakes" or" confirms". When a backup-archive client is sending data to the server, confirms are generated and sent from the client. ADSM performs this handshaking at the application level. The confirm is like a client heartbeat during a session while data is being transferred.

3.2.3.2 Version 2 Confirms

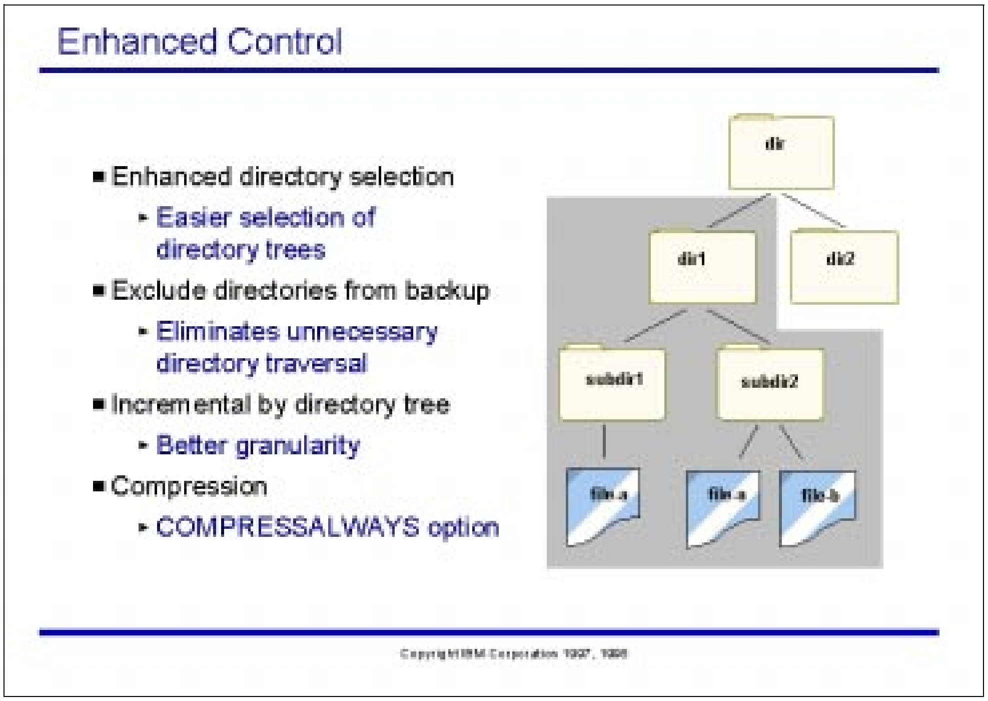
The Version 2 implementation sends confirms based on the size of the file. The confirms are also tied to file level boundaries, so that confirms are always performed at the end of each file. The graphic shows three files, file a, b, and c. Packets of data are sent from the client to the server in addition to periodic confirms. In Version 2, confirms are performed after a predetermined amount of data has been transferred and at the end of each file.

3.2.3.3 Version 3 Confirms

The new process performs confirms based on time and is tied to transaction boundaries. The graphic shows the same three files in a Version 3 environment. After a predetermined amount of time, the client generates a confirm. This allows for a client to better account for a fast network. In Version 2, the amount of data dictated when a confirm would be sent. In a faster network, this would result in confirms being sent more frequently than needed. Because the confirm generation is based on time in Version 3, fewer confirms are needed to accomplish the same goal. Confirms are sent at transaction boundaries, rather than file boundaries, which also helps to reduce the number of confirms sent to the server.

3.2.3.4 Benefit

This new confirmation process allows confirms to be sent more efficiently and intelligently, reducing the overhead for transactions, which in turn provides better overall performance. This is also consistent with basing events on transactions, which is a unit of work committed to the ADSM database. (Server aggregation bases the grouping of files on a transaction. The restartable restore resumes a restore at a transaction boundary.)



Enhanced Control

- Enhanced directory selection
  - Easier selection of directory trees
- Exclude directories from backup
  - Eliminates unnecessary directory traversal
- Incremental by directory tree
  - Better granularity
- Compression
  - COMPRESSALWAYS option

dir
dir1
dir2
subdir1
subdir2
file.a
file.a
file.b

Copyright IBM Corporation 1997, 1998

3.2.4 Enhanced Control

Directory handling is enhanced to provide more granularity and better control of client filespaces during incremental backups. This enhanced control of filespaces can reduce the amount of data being backed up, thereby reducing overall elapsed time. A new client compression option has been provided that reduces the number of client retries resulting from compression that occur with Version 2.

3.2.4.1 Enhanced Directory Selection

By improving the directory selection process, desired data can be more easily and accurately selected for transfer.

3.2.4.2 Excluding Directories from Backup

Directories can be specifically excluded from being traversed during an incremental backup, resulting in a more efficient session and reducing time to complete a backup. Eliminating specified directory trees from being traversed provides a performance improvement over Version 2 processing. A larger exclusion of directory paths would result in a greater performance benefit.

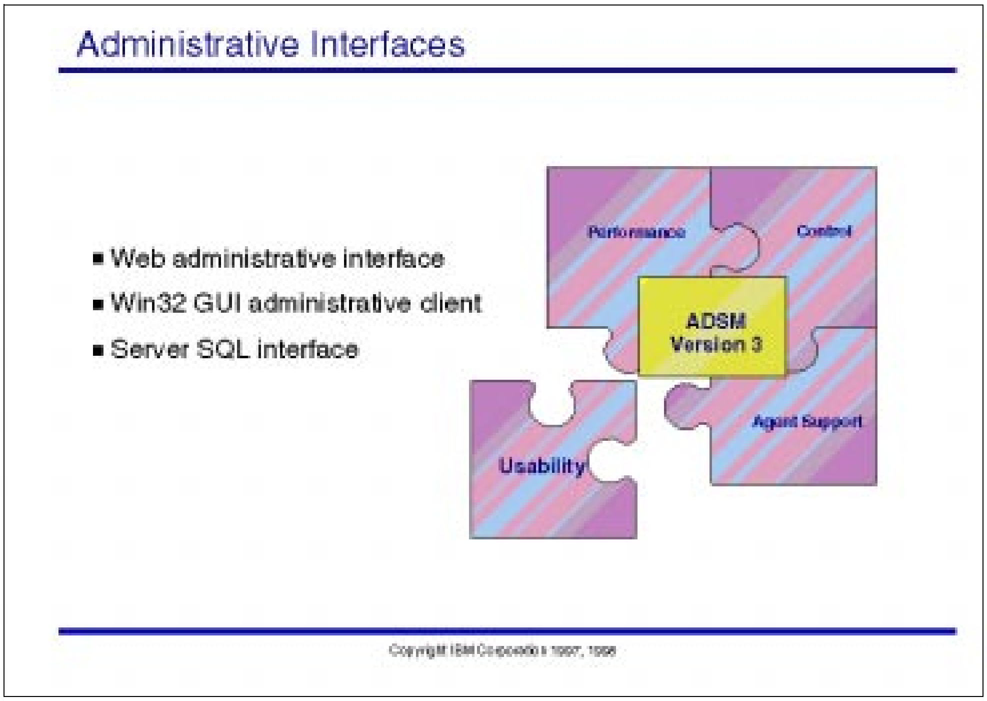
3.2.4.3 Incremental by Directory Tree

Incremental backups by directory tree allow more granularity and reduce the amount of data needed to be transferred in some cases. The overall time to complete a task is thus reduced, resulting in faster execution than the Version 2 process, which provides granularity only to the filespace level.

3.2.4.4 Compression

The backup-archive client can detect when files grow as a result of being compressed during a backup operation. Typically, this occurs when files have previously been compressed by some other application. With Version 2 clients, this results in the transaction being terminated and the files being resent without client compression. However, in some cases, it is faster to allow the process to continue rather than to abort the transaction and resend the files uncompressed. A new Version 3 client option, COMPRESSALWAYS, enables this and allows such transactions to proceed.

Chapter 4. Administrative Interfaces



Administrative Interfaces

- Web administrative interface
- Win32 GUI administrative client
- Server SQL interface

Performance
Control
ADSM
Version 3
Agent Support
Usability

Copyright IBM Corporation 1997, 1998

This chapter examines the Version 3 administrative interfaces. It covers the following topics:

- Web administrative interface

  Version 3 introduces a Web interface for administration. This section describes this new interface and its configuration.

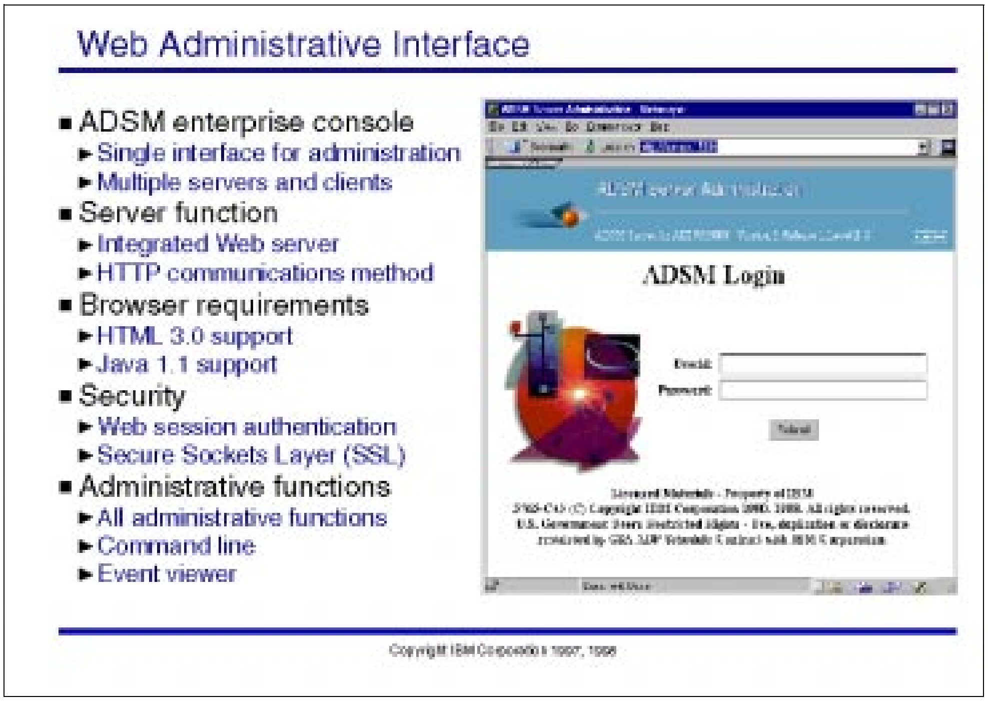
- Win32 GUI administrative client

  Version 3 provides only one GUI administrative client, an enhanced Windows 32-bit client for Windows NT and Windows 95.

- Server SQL interface

  Version 3 introduces a structured query language (SQL) interface that you can use to query server information. There are two methods of using the SQL interface: the command line administrative client and a new open database connectivity (ODBC) driver.

4.1 Web Administrative Interface



Web Administrative Interface

- ADSM enterprise console
  - Single interface for administration
  - Multiple servers and clients
- Server function
  - Integrated Web server
  - HTTP communications method
- Browser requirements
  - HTML 3.0 support
  - Java 1.1 support
- Security
  - Web session authentication
  - Secure Sockets Layer (SSL)
- Administrative functions
  - All administrative functions
  - Command line
  - Event viewer

ADSM Server Administration
ADSM Login
Userid:
Password:
Submit
Licensed Materials - Property of IBM
5765-C43 (C) Copyright IBM Corporation 1990, 1998. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corporation.

Copyright IBM Corporation 1997, 1998

In ADSM Version 3 the Web administrative interface is the primary interface for administration. It incorporates Java applets and provides the following functions:

- ADSM enterprise console

  The enterprise console displays ADSM as a single entity in the enterprise. It is an interface that integrates ADSM server and client functions for the administrator as a single application to manage ADSM in a distributed environment

- Server function

  The Web administrative interface is an ADSM server function and consists of an integrated Web server and a new HTTP server communications protocol.

- Browser requirements

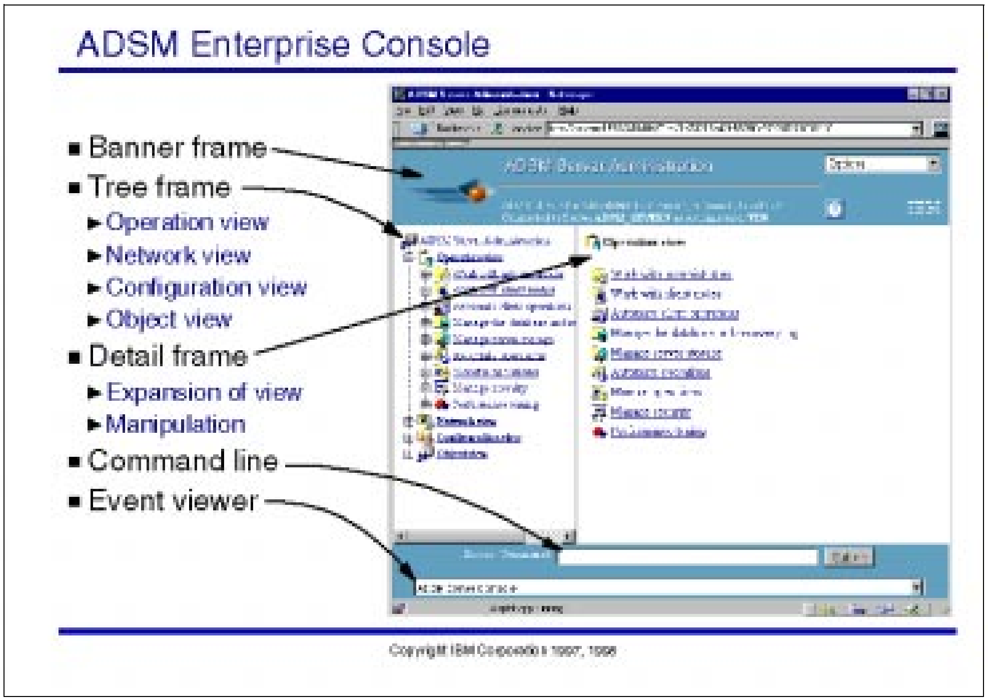
  The Web administrative interface can be used from any workstation running an HTML 3.0 compliant Web browser with support for Java 1.1.5 or higher. Netscape 4.03, 4.04, and 4.05 requires a JDK 1.1.5 upgrade for this support. Netscape 4.06 has the required Java support. Microsoft’s Internet Explorer 4.01 has the required Java support.

- Security

  Security for the Web administrative interface is based on enhanced logging of Web-based sessions, password restrictions, and administrator lockout capabilities. In addition Secure Sockets Layer (SSL) communications has been introduced in Version 3.1.2.

- Administrative functions

The Web administrative interface can be used to perform all administrative functions graphically, and it provides a command line and server event viewer.



ADSM Enterprise Console

- Banner frame
- Tree frame
  - ▸ Operation view
  - ▸ Network view
  - ▸ Configuration view
  - ▸ Object view
- Detail frame
  - ▸ Expansion of view
  - ▸ Manipulation
- Command line
- Event viewer

Copyright IBM Corporation 1997, 1998

4.1.1 ADSM Enterprise Console

When an administrator logs in to the server through the Web administrative interface, the ADSM enterprise console is displayed. The enterprise console consists of three frames. Additional command line and event viewers can be displayed.

4.1.1.1 Banner Frame

The banner frame indicates which server you are connected to (server name, platform, and version, release, level) and the fact that you are using the ADSM enterprise console. It also indicates the name you are logged on as administrator.

The only operable control on the banner frame is a selection button that you can use to choose whether you want to display the command line and the event viewer. You can turn those windows off or on depending on what you want to see in your browser display. Another option offers you the ability to log off (forcing reauthentication).

4.1.1.2 Tree Frame

A collapsible tree is used for navigation. The content and layout are determined by the “view” that you have chosen. The initial branches of the tree are the views from which you can choose. Views are simply paradigms that are used to navigate to the objects in the enterprise that you want to use. Available views are:

- The operation view, which when expanded displays a navigation tree that consists of all administrative functions of the Web administrative interface.
- The network view, which when expanded displays a navigation tree that consists of server groups, servers, clients, and the command routing function of the interface.

- The configuration view, which when expanded displays a navigation tree of the configuration settings, the configuration profiles, and the profile subscriptions that have been established on the configuration management server.
- The object view, which when expanded displays a navigation tree of all groups and classes in the basic hierarchy.

4.1.1.3 Detail Frame

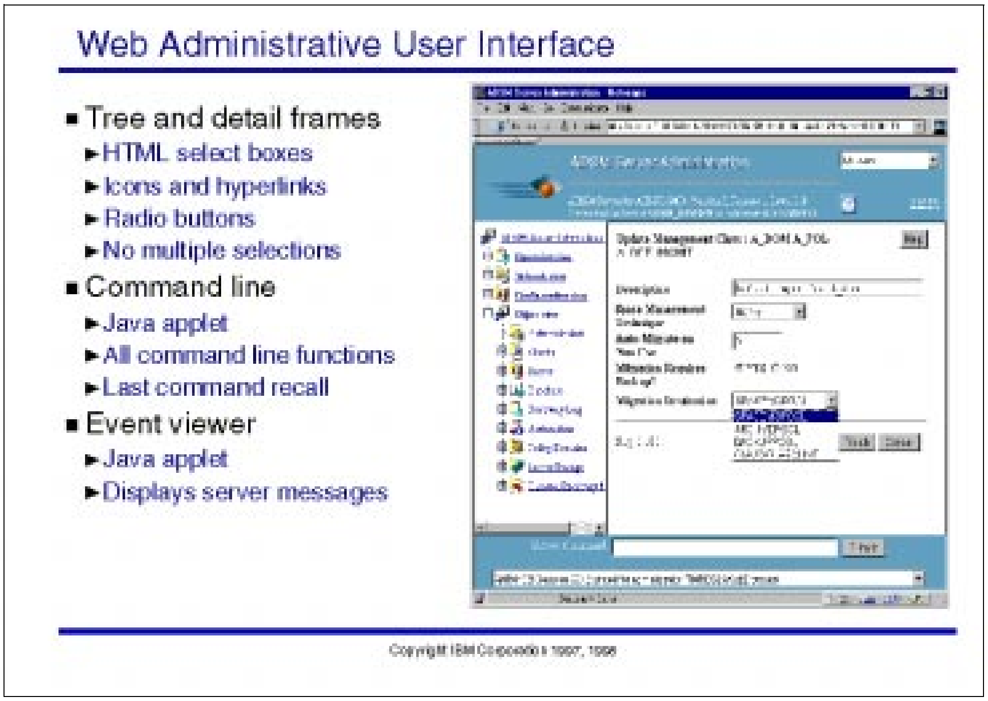
The detail frame displays detailed information about the item that you have selected from the tree frame. All operations that are available to manipulate the selected item are provided in the form of a pull-down menu in the top-right corner of the detail frame. For example, when a server group is selected, its attributes are displayed in the detail frame. Operations are provided for adding servers to the group, removing servers from the group, and defining new server groups.

4.1.1.4 Command Line

An administrative command line can be displayed by selecting it from the selections pull-down menu in the banner frame.

4.1.1.5 Event Viewer

A server event viewer can be displayed by selecting it from the selections pull-down menu in the banner frame.



Web Administrative User Interface

- Tree and detail frames
  - HTML select boxes
  - Icons and hyperlinks
  - Radio buttons
  - No multiple selections
- Command line
  - Java applet
  - All command line functions
  - Last command recall
- Event viewer
  - Java applet
  - Displays server messages

4.1.2 Web Administrative User Interface

The Web administrative user interface consists of three interfaces integrated as one client. The tree and detail frames provide a graphical interface for performing administrative functions. Java applets provide administrative command line and event viewer functions.

4.1.2.1 Tree and Detail Frames

The tree and detail frames of the Web administrative user interface use HTML features to provide an easy to use graphical interface:

- Select boxes to enable users to select possible options from a list
- Icons and hyperlinks to enable users to navigate through the interface by using the mouse to point and click on icons and keywords
- Radio buttons that present the available options on screen and enable users to click with the mouse to select items

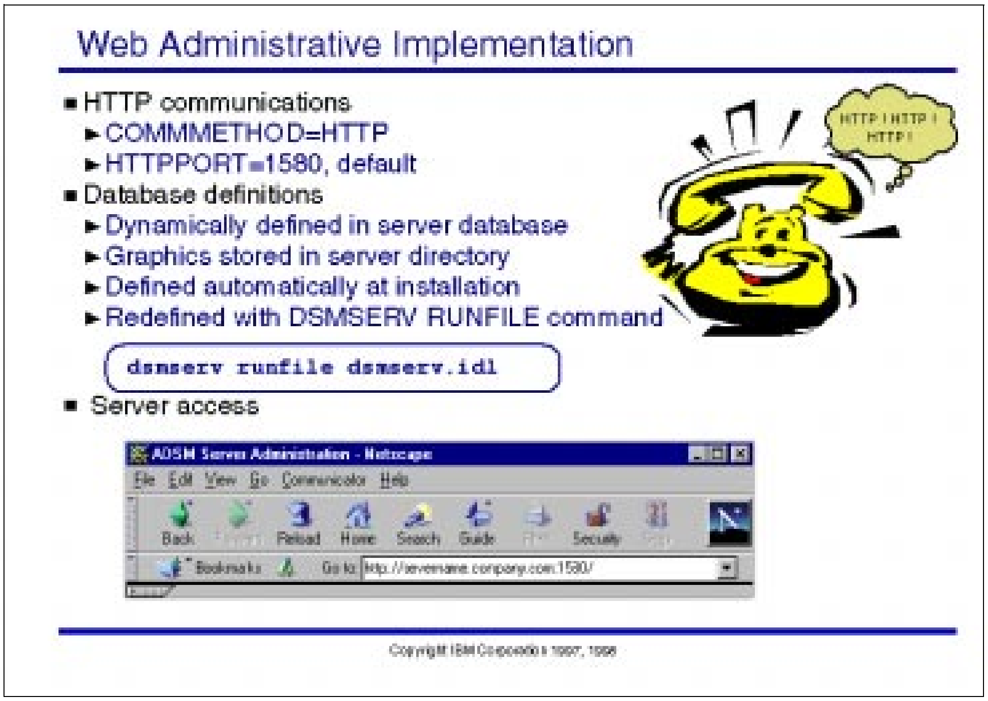
Multiple objects cannot be selected at once. Only a single object at a time can be selected for operations through the Web administrative user interface.

4.1.2.2 Command Line

The command line is a Java applet that consists of a single text input control and submit button. Administrative commands can be entered, and clicking on the submit button displays the results in the detail frame. All administrative commands can be issued from the command line. The command line can display commands previously entered so that they can be recalled, modified, and resubmitted.

4.1.2.3 Event Viewer

The event viewer is a Java applet implemented as a pull-down selection box. A single line of text is displayed in the box when it is collapsed. The text consists of the latest messages issued to the server console. You can view the last messages by pulling down the selection box control and scrolling forward and backward through the messages. The event viewer is read-only.



Web Administrative Implementation

- HTTP communications
  - COMMMETHOD=HTTP
  - HTTPPORT=1580, default
- Database definitions
  - Dynamically defined in server database
  - Graphics stored in server directory
  - Defined automatically at installation
  - Redefined with DSMSERV RUNFILE command

```
dsmserv runfile dsmserv.idl
```

- Server access

HTTP ! HTTP ! HTTP !
ADSM Server Administration - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks
Go to: http://servername.company.com:1580/

Copyright IBM Corporation 1997, 1998

4.1.3 Web Administrative Implementation

The Web administrative interface is installed and configured automatically at server installation time. However, use of the interface is optional and can be disabled for environments where it is not required.

4.1.3.1 HTTP Communications

A new HTTP communications protocol is provided for the ADSM server. Defining this protocol starts an internal HTTP daemon (Web server) when the ADSM server is started.

Two new options have been added to the dsmserv.opt file to configure the Web administrative interface. A new value for the COMMETHOD option is HTTP, which can be used in conjunction with other values of the COMMETHOD option. The COMMETHOD=HTTP option starts the Web administrative daemon when the ADSM server is started. A second new server option, HTTPPORT, specifies the TCP/IP port that the Web server interface monitors for incoming HTTP connections. It has a default value of 1580 but can be configured to other values if required.

4.1.3.2 Database Definitions

To provide flexibility for adding future functions or enhancements, the HTML code presented to the administrator’s Web browser is not hard coded within the ADSM database. The server HTTP daemon accepts requests from the administrator's Web browser in the form of a URL and translates them into instructions for the server through the use of object classes, operations, and groupings stored within the server database. The server processes the commands and returns any output to the HTTP daemon, which translates the text into HTML and transfers it to the administrator’s Web browser.

The graphics displayed as icons in the Web administrative interface are stored in the ADSM server install directory. The icons are stored in GIF 89a format and are interlaced to improve speed and portability of the interface.

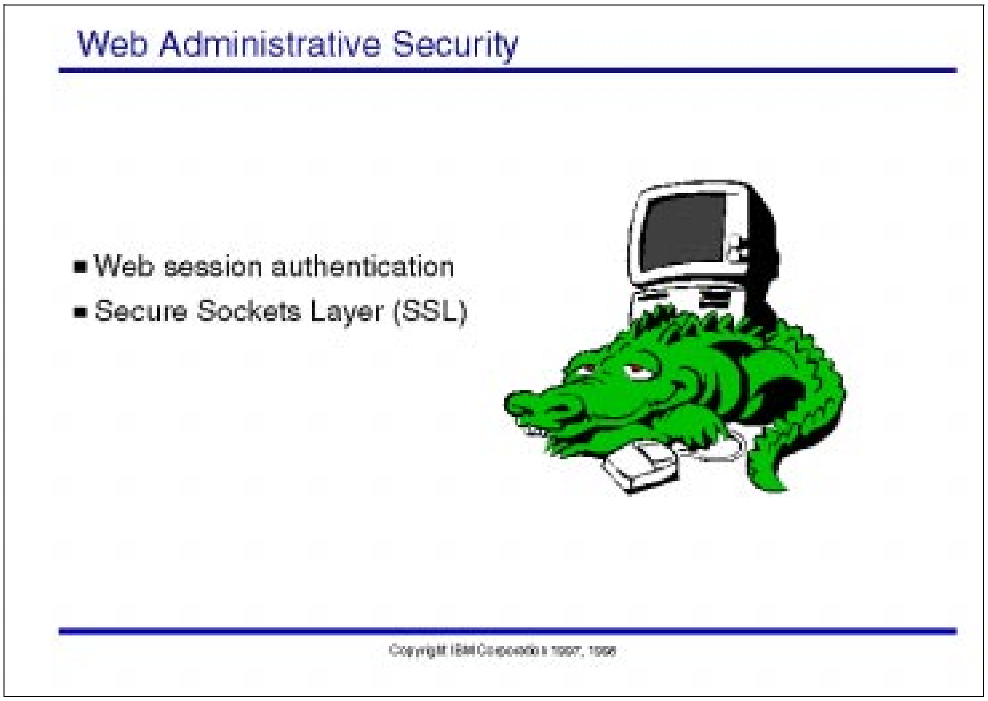
To configure the Web administrative interface automatically during the server installation process, you run an interface definition job, which creates all of the required database definitions. If the server is reinstalled, you must reconfigure the Web administrative interface, using the RUNFILE option with the DSMSERV executable. This reconfiguration re-creates the HTML definitions for all server objects in the database and enables the Web administrative interface.

4.1.3.3 Server Access

To access the Web administrative interface, you specify the server name and the port number (1580 by default) in the Web browser’s location text field:

```
http://sever.company.com:1580
```

You are then connected to the server and prompted for an administrative ID and password. Once you have entered the correct ID and password, the Web administrative interface is displayed in the browser.



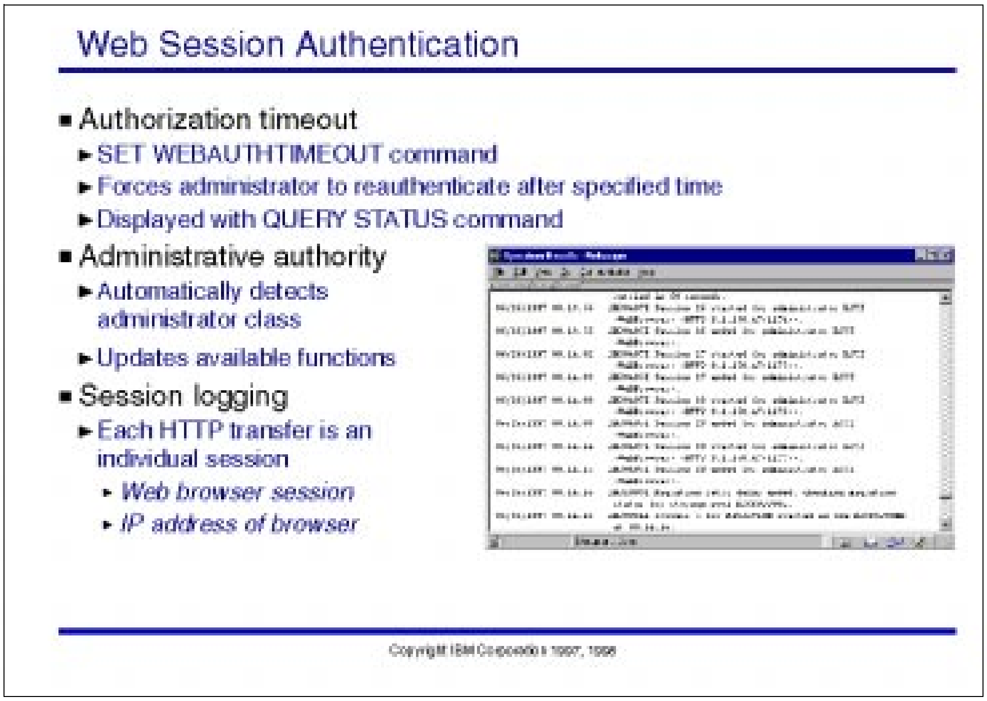
Web Administrative Security

- Web session authentication
- Secure Sockets Layer (SSL)

4.1.4 Web Administrative Security

Various enhancements have been made to the security of Version 3. This section covers the following topics:

- Web Session Authentication
- Secure Sockets Layer (SSL)



Web Session Authentication

- Authorization timeout
  - SET WEBAUTHTIMEOUT command
  - Forces administrator to reauthenticate after specified time
  - Displayed with QUERY STATUS command
- Administrative authority
  - Automatically detects administrator class
  - Updates available functions
- Session logging
  - Each HTTP transfer is an individual session
    - *Web browser session*
    - *IP address of browser*

Copyright IBM Corporation 1997, 1998

4.1.5 Web Session Authentication

The main changes that support the Web administrative interface are authorization timeouts, administrative authority, and session logging.

4.1.5.1 Authorization Timeout

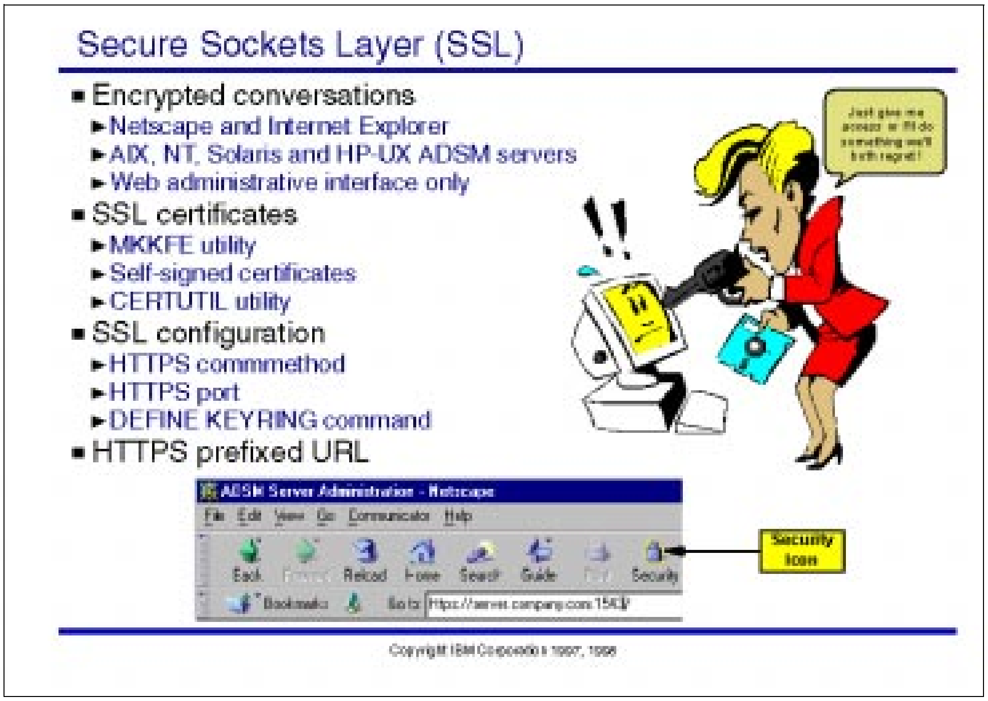
To avoid the situation whereby an unauthorized person could use a cached copy of the Web administrator interface, a new timeout parameter can be specified at the server. SET WEBAUTHTIMEOUT forces an administrator to revalidate his or her administrative ID and password after a specified period of time. This value can be displayed with the QUERY STATUS command.

4.1.5.2 Administrative Authority

Because of the way in which the HTTP daemon interfaces with the server database, the Web administrative interface can recognize the privileges of the administrator logging in and adapt the available options to suit his or her assigned authority. Thus administrators are prevented from attempting to perform functions for which they do not have the required authority.

4.1.5.3 Session Logging

Each transfer of information from the ADSM server to the Web administrative interface is logged as a separate session because HTTP is a stateless protocol. Stateless protocols operate such that the client connects, makes a single request, gets a single response, and then disconnects. This process is then repeated for each transfer between the browser and the HTTP interface. Sessions are logged in the ADSM activity log as a “Web-Browser Session” along with the IP address of the machine being used to run the browser. Thus the administrator can easily identify any Web administrative sessions and determine from which machine the browser is being run.



Secure Sockets Layer (SSL)

- Encrypted conversations
  - Netscape and Internet Explorer
  - AIX, NT, Solaris and HP-UX ADSM servers
  - Web administrative interface only
- SSL certificates
  - MKKFE utility
  - Self-signed certificates
  - CERTUTIL utility
- SSL configuration
  - HTTPS commmethod
  - HTTPS port
  - DEFINE KEYRING command
- HTTPS prefixed URL

Just give me access or I'll do something we'll both regret!
!!
ADSM Server Administration - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security
Bookmarks
Go to: https://server.company.com:1543/
Security Icon

Copyright IBM Corporation 1997, 1998

4.1.6 Secure Sockets Layer (SSL)

ADSM Version 3.1.2 provides support for Secure Sockets Layer (SSL) conversations between the ADSM server and an administrator's Web browser.

4.1.6.1 Encrypted Conversations

SSL provides secure, encrypted conversations between Web browsers and Web servers. With SSL, all TCP/IP send and receive data transfers are passed through an SSL API. Thus all data is encrypted before it is sent over the network, and it is decrypted when received from the network.

The Netscape Navigator and Internet Explorer browsers support SSL and can be used to administer ADSM Version 3.1.2 servers running on AIX, Windows NT, Solaris, and HP-UX. The MVS server does not currently support SSL.

ADSM Version 3 only supports SSL for the Web administrative interface. Backup/Archive client sessions do not use SSL encryption.

4.1.6.2 SSL Certificates

The SSL protocol is based on certificates. When a Web browser first connects to the ADSM server, with SSL enabled, the server sends a certificate to the browser, which must return a matching certificate before the session is established. The encryption key for the subsequent data transfers is held within the certificate. Before SSL can be enabled on the ADSM server, an SSL certificate must be generated and signed.

SSL certificates are held in a *keyring file,* which contains one or more application specific *certificates*, or *keys*. ADSM Version 3 provides a utility, MKKFE, to create a keyring file and certificate for the ADSM server. The ADSM server keyring file is created first, followed by a certificate. The certificate is created by generating a *certificate request* for the ADSM server. This certificate request can then be *signed* by a trusted Certification Authority (CA). Signing the certificate authenticates it for use. Certification Authorities are typically telecommunications companies that charge a fee for providing this service. MKKFE is then used to receive the signed certificate into the ADSM server keyring file.

ADSM provides another utility, CERTUTIL, that allows you to become your own Certification Authority. CERTUTIL can be used to sign the certificate request, without the need to use an external Certification Authority. CERTUTIL is provided only with ADSM Version 3 for Windows NT. It is not available for the other server platforms.

Self-Signed Certificates

Certificate requests generated with MKKFE can be received into the keyring file without being signed by a trusted Certification Authority. These are termed *self-signed certificates*.

Netscape Navigator supports the use of self-signed certificates. Navigator displays a warning message that a self-signed certificate is being used when an administrator first accesses the ADSM server. Subsequent warning messages can be disabled, and SSL will function correctly.

Internet Explorer does not support self-signed certificates and will not work with them. If Internet Explorer is to be used, the certificate must be signed, either through the CERTUTIL utility or by a trusted Certification Authority.

4.1.6.3 SSL Configuration

SSL is a new communications method for the ADSM server. It is configured with a new option, COMMETHOD HTTPS, in the server options file. Associated with this communications method is an associated port number on which the server listens for SSL conversations. This port number is defined with the new HTTPSPORT option in the server options file. If not defined, the port number defaults to 1543.

The keyring file generated for the ADSM server is password protected for added security. The keyring file name and its password must be defined to the ADSM server through the new DEFINE KEYRING command:

```
adsm> define keyring "keyringfilename" "keyring_password"
```

After you define the keyring filename and password, update the server options file, and restart the server, the HTTPS communication method will start. The server activity log contains a message indicating that HTTPS is being initialized and the port number being monitored for SSL connections:

```
ANR8282I HTTPS driver ready for connection with clients on port 1543.
```

An improperly generated or corrupted certificate or an invalid password prevents the HTTPS protocol from starting. Errors such as these are also displayed in the server activity log.

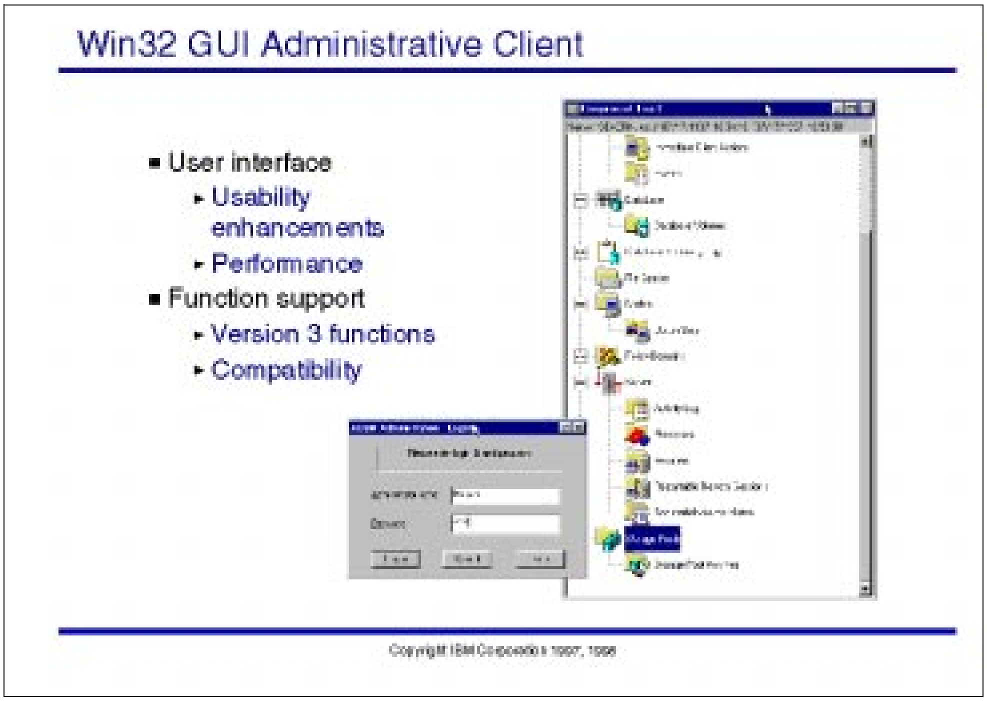
4.1.6.4 HTTPS Prefixed URL

Having configured SSL on the ADSM server, an administrator can then connect to the server in the normal manner, but the ADSM server URL must be prefixed with HTTPS rather than HTTP. Failure to use HTTPS causes the browser to hang.

If a self-signed certificate is being used, the browser issues a warning message to this effect. These warning messages can be disabled from being displayed for subsequent sessions.

When Netscape invokes a secured session it displays a different security icon (a closed rather than open padlock) on the menu bar. Clicking on this padlock icon displays details of the security information including details of the ADSM server certificate.

4.2 Win32 GUI Administrative Client



Win32 GUI Administrative Client

- User interface
  - Usability enhancements
  - Performance
- Function support
  - Version 3 functions
  - Compatibility

Copyright IBM Corporation 1997, 1998

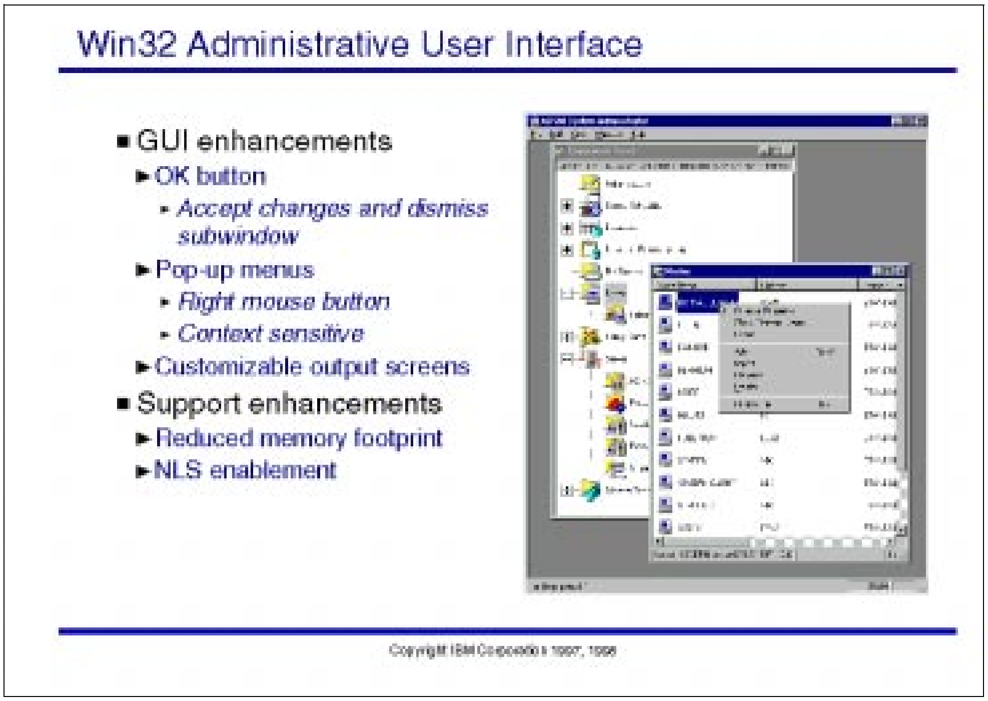
In Version 3 only one administrative GUI client is provided. This is for Windows NT and Windows 95. As the only GUI administrative client for Version 3, the Win32 interface has been updated:

- User interface

  A number of overall improvements have been made to both the performance and the usability of the Win32 administrative GUI. These include reduced memory footprint, context-sensitive menus, and other advances in interface technology.

- Function support

  In line with the Web administrative interface, the number of ADSM Version 2 functions now supported by the Win32 administrative GUI has also increased. Thus the Win32 administrative GUI is more complete in terms of function, and the ADSM administrator can perform more of the ADSM functions graphically.



Win32 Administrative User Interface

- GUI enhancements
  - OK button
    - *Accept changes and dismiss subwindow*
  - Pop-up menus
    - *Right mouse button*
    - *Context sensitive*
  - Customizable output screens
- Support enhancements
  - Reduced memory footprint
  - NLS enablement

4.2.1 Win32 Administrative User Interface

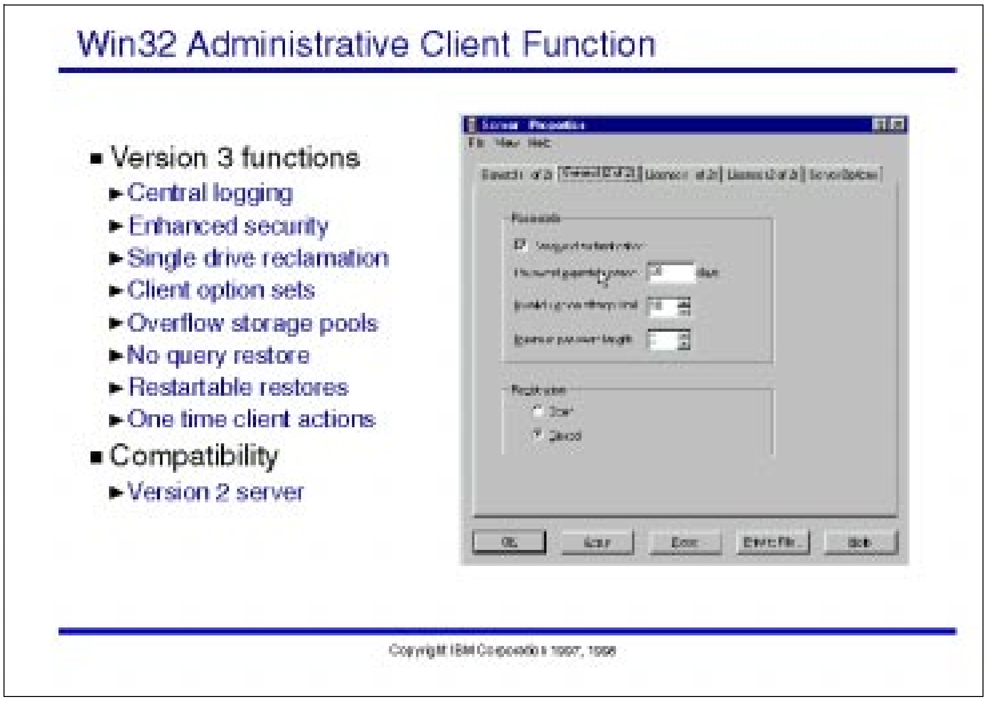
The Win32 administrative GUI has had a number of modifications both in the way the client interfaces with the operating system and the user interfaces with the client. These modifications have led to improvements in performance and usability of the client interface.

4.2.1.1 GUI Enhancements

Since the release of Version 2, there have been a number of improvements in GUI design and requests for changes from ADSM users. Many of these improvements have been included in the Win32 administrative GUI for Version 3. OK buttons have been added to many of the dialogs that accept the changes that have been made and close the window in one action. More use has been made of the right mouse button. Clicking in certain areas of the screen now produces a context-sensitive menu with items pertaining to that area of the screen. In the settings screens for various ADSM objects, the display columns allow customization of size and location. It is also possible to hide certain columns completely.

4.2.1.2 Support Enhancements

The memory footprint of the Win32 administrative client has been reduced to lower the amount of memory required on the client workstation. The Win32 administrative GUI now has single-byte and multibyte character set support, so it can support more languages.



Win32 Administrative Client Function

- Version 3 functions
  - Central logging
  - Enhanced security
  - Single drive reclamation
  - Client option sets
  - Overflow storage pools
  - No query restore
  - Restartable restores
  - One time client actions
- Compatibility
  - Version 2 server

Copyright IBM Corporation 1997, 1998

4.2.2 Win32 Administrative Client Function

The Win32 administrative client has been updated to support new Version 3 functions.

4.2.2.1 Version 3 Functions

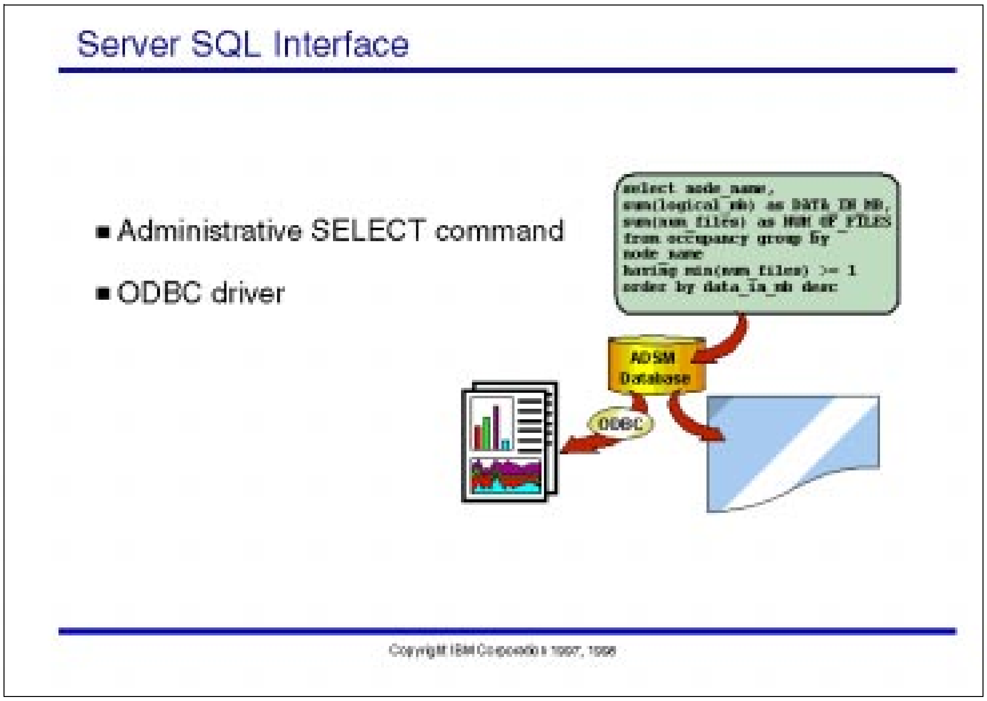
Many of the new functions within Version 3 have been externalized in the Win32 administrative GUI. The specific functions supported include:

- Central logging - configuration of events to be sent to the server and processed by various monitors. Accessed through the FILE menu in the server properties dialog
- Enhanced security - the ability for the administrator to set password restrictions and force client password reset
- Single drive reclamation - process to enable reclamation to be run on a single drive library
- Client option sets - client options stored on the server that can be allocated to clients when they are defined
- Overflow storage pools - new location for volumes in storage pools that become too large to fit into a single library
- No query restore - enhanced restore protocol to reduce memory usage on the client and improve performance of restores
- Restartable restores - no query restore sessions that have been interrupted and may be restarted
- One time client actions - schedules defined with the DEFINE CLIENTACTION command for immediate processing and automatic deletion from the ADSM database

4.2.2.2 Compatibility

The Win32 administrative client can be used to administer a save Version 2 server. The Version 3 specific functions do not appear on the GUI screen.

4.3 Server SQL Interface



Server SQL Interface

- Administrative SELECT command
- ODBC driver

```
select node_name,
sum(logical_mb) as DATA_IN_MB,
sum(num_files) as NUM_OF_FILES
from occupancy group by
node_name
having min(num_files) >= 1
order by data_in_mb desc
```

Copyright IBM Corporation 1997, 1998

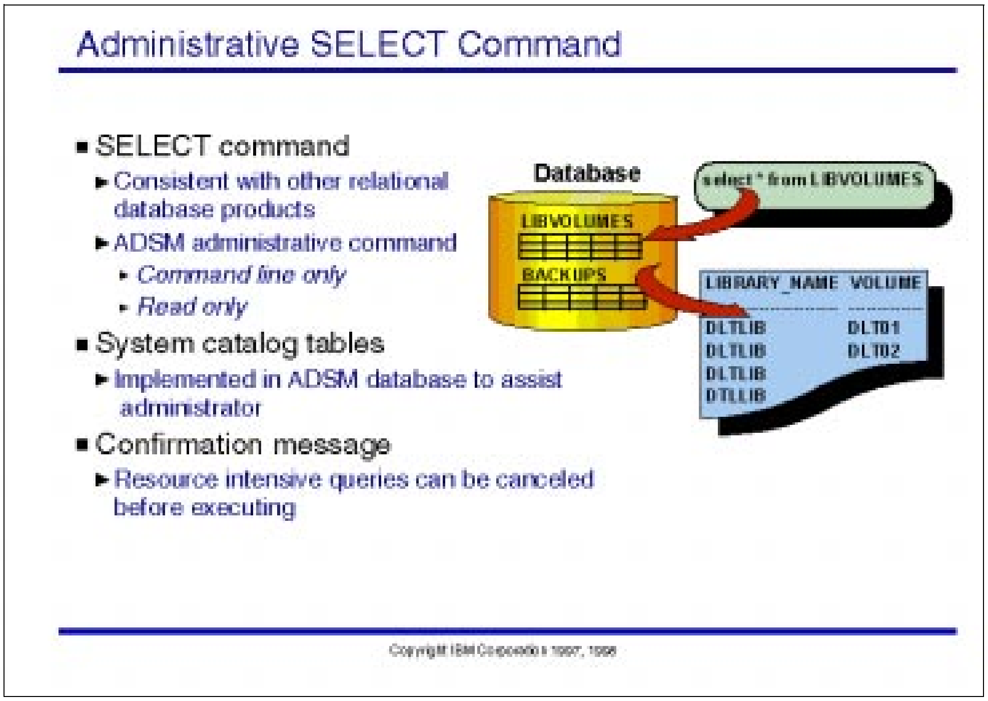
This section covers the new ADSM server SQL interface introduced with Version 3. The following topics are covered:

- Administrative SELECT command

  A new administrative command, SELECT, has been introduced. An administrator can use the SELECT command to query the ADSM server database.

- ODBC driver

  An ODBC driver is also provided with Version 3. With this driver, desktop database products such as Lotus Approach or Microsoft Access can be connected to the ADSM server database. Thus administrators can use these tools to perform ADSM server queries in an easy-to-use, graphical manner.



Administrative SELECT Command

- SELECT command
  - Consistent with other relational database products
  - ADSM administrative command
    - *Command line only*
    - *Read only*
- System catalog tables
  - Implemented in ADSM database to assist administrator
- Confirmation message
  - Resource intensive queries can be canceled before executing

Database
LIBVOLUMES
BACKUPS

```
select * from LIBVOLUMES
```

```
LIBRARY_NAME  VOLUME
------------  ------
DLTLIB        DLT01
DLTLIB        DLT02
DLTLIB
DTLLIB
```

Copyright IBM Corporation 1997, 1998

4.3.1 Administrative SELECT Command

The ADSM database stores information about the storage management policy, client data objects (directories and files), and storage hierarchy. With Version 2 this information was available only through a set of ADSM query commands that produced formatted output; there was little opportunity to produce customized output. Version 3 addresses this requirement by providing an SQL interface.

4.3.1.1 SELECT Command

The SQL interface consists of an SQL SELECT command for the ADSM server. The interface is consistent with relational database products and presents server information in the form of relational tables containing rows and columns. Select statements can be issued from:

- ADSM administrative command line interface
- ADSM Web administrative interface

The interface is read only and allows queries on the ADSM database. The contents or structure of the ADSM database cannot be changed. Therefore stored procedures cannot be held in the ADSM database. A stored procedure is a predefined SQL statement similar to an ADSM macro.

4.3.1.2 System Catalog Tables

Three system catalog tables are implemented to assist the administrator in determining the information available:

SYSCAT.TABLES Contains information about all tables that are available for querying with the select statement

SYSCAT.COLUMNS Describes the columns that reside in each of the tables

SYSCAT.ENUMTYPES For columns that have an enumerated data type, defines the legal values for each enumerated data type and the ordering of the different values for the type. An enumerated data type is a value that is assigned a numerical number rather than text.

The following select statements are examples of querying information from the system catalog tables:

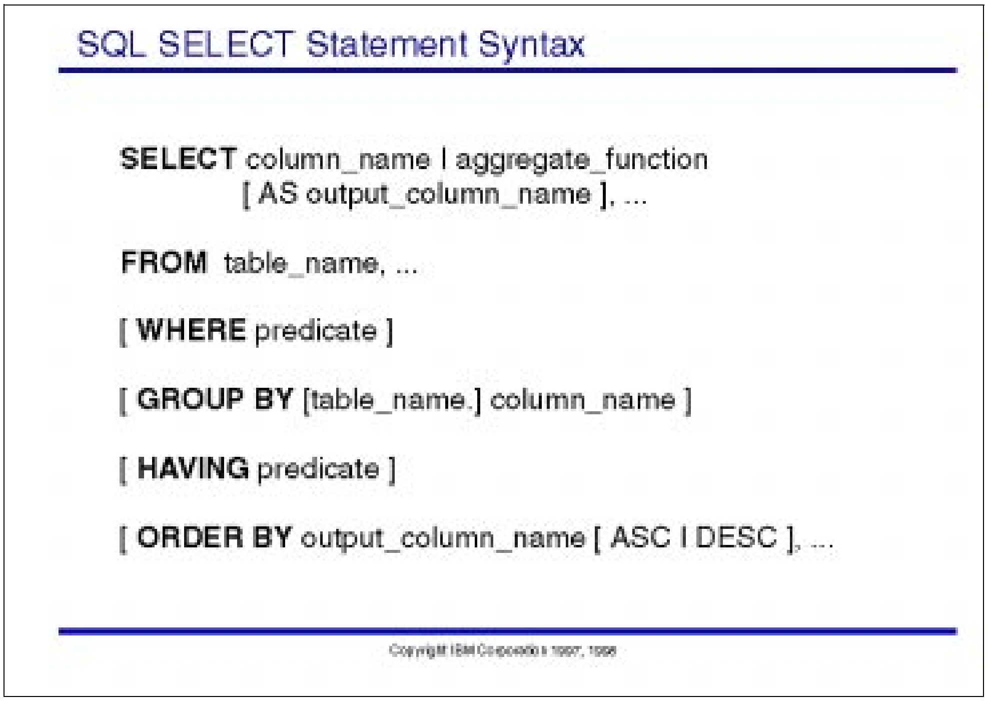
```
adsm> select tabschema, tabname, unique_index from syscat.tables

TABSCHEMA     TABNAME                UNIQUE_INDEX
---------     ------------------     ------------
ADSM          ACTLOG                        FALSE
ADSM          ADMINS                         TRUE
ADSM          ADMIN_SCHEDULES                TRUE
ADSM          ARCHIVES                      FALSE
...
```

```
adsm> select tabname, colname, typename from syscat.columns

TABNAME           COLNAME                TYPENAME
-------------     ------------------     -------------------------
ACTLOG            DATE_TIME              TIMESTAMP
ACTLOG            MSGNO                  INTEGER
ACTLOG            SEVERITY               ENUMERATED(SEVERITY_TYPE)
ACTLOG            MESSAGE                VARCHAR
ACTLOG            ORIGINATOR             VARCHAR
...
```

```
adsm> select typename, values from syscat.enumtypes

TYPENAME           VALUES
--------------     --------------------------------------------------
OBJECT_TYPE        DIR(0), FILE(1), UNKNOWN(2)
BACKUP_STATE        ACTIVE_VERSION(0), INACTIVE_VERSION(1), UNKNOWN(2)
YESNO_TYPE         NO(0), YES(1)
COMPRESSTYPE       NO(0), YES(1), CLIENT(2)
OPENCLOSED         CLOSED(0), OPEN(1)
LOGGINGMODE        NORMAL(0), ROLLFORWARD(1)
SEVERITY_TYPE      I(0), W(1), E(2), S(3), D(4)
```

4.3.1.3 Confirmation Message

For SQL queries that operate on large tables or require significant time and resources from the ADSM server, a confirmation message is displayed indicating that the query is resource intensive and may take a lot of time to generate results. This message provides the opportunity to abort the query before resources are consumed or to run it anyway. Because queries may tie up sessions for some time, they cannot be executed from the server console. The console session is thus available for managing other administrative functions.



SQL SELECT Statement Syntax

```
SELECT column_name | aggregate_function
          [ AS output_column_name ], ...

FROM  table_name, ...

[ WHERE predicate ]

[ GROUP BY [table_name.] column_name ]

[ HAVING predicate ]

[ ORDER BY output_column_name [ ASC | DESC ], ...
```

Copyright IBM Corporation 1997, 1998

4.3.2 SQL SELECT Statement Syntax

To use the SQL interface a basic understanding of the SQL SELECT statement is required. The syntax of the ADSM SELECT statement is compliant with ANSI SQL SQL92/93 syntax but has some limitations. These limitations reduce query complexity and the corresponding load on the ADSM server database.

An SQL select statement can be divided into a number of clauses.

4.3.2.1 SELECT and FROM

The basic clause is SELECT, used to select columns FROM tables. The columns selected can optionally be aggregated and displayed with a different output column name:

column_name column name of the table or tables specified in the FROM clause

aggregate_function functions such as SUM, COUNT, MAX, or AVG that extract a single value from a group of columns

output_column_name column title displayed by the query output for the specified column name (default is to display the original column name)

table_name table name from which query rows and columns are extracted

The simplest form of a select statement is to SELECT all columns FROM a table; the following example selects all rows from the SESSIONS table:

```
adsm> select * from sessions

     SESSION_ID: 1126
     START_TIME: 1997-08-19 10:48:36.000000
     COMMMETHOD: Tcp/Ip
          STATE: Run
   WAIT_SECONDS: 0
     BYTES_SENT: 2301
 BYTES_RECEIVED: 175
   SESSION_TYPE: Administrative
CLIENT_PLATFORM: WinNT
    CLIENT_NAME: TIM
     OWNER_NAME:
    MEDIA_STATE:
```

Note

The SESSIONS table is a dynamic table created and updated as sessions start and stop.

4.3.2.2 WHERE

As part of a SELECT statement the WHERE clause can be used to predicate, or limit, the rows returned from the query:

predicate expression to limit rows returned from the query:

- node_name='POLONIUM'
- logical_mb>=1000
- platform_name IS NULL
- bytes_sent BETWEEN 100 AND 500
- nodes.node_name=occupancy.node_name (join criteria for selecting one value from two tables)

The following statement, an example of using a simple where clause, selects all client schedules that are run on a Sunday:

```
adsm> select schedule_name, action, starttime from client_schedules
       where dayofweek= 'SUNDAY'

SCHEDULE_NAME          ACTION                 STARTTIME
------------------     ------------------     ---------
WEEKLY                 SELECTIVE               01:00:00
```

The statement below is example of using join criteria. The join criteria are used to avoid redundant output lines if more than one table is queried in a SELECT statement. The statement provides information about client schedules and the associated client nodes:

```
adsm> select node_name, client_schedules.schedule_name, dayofweek,
       starttime from associations, client_schedules where
        associations.schedule_name = client_schedules.schedule_name

NODE_NAME           SCHEDULE_NAME         DAYOFWEEK         STARTTIME
---------------     ------------------    --------------    ---------
DANUBE              DAILY_INC             ANY                19:00:00
SEVERN              DAILY_INC             ANY                19:00:00
NOBELIUM            DAILY_INC             ANY                19:00:00
DANUBE              WEEKLY                SUNDAY             01:00:00
SEVERN              WEEKLY                SUNDAY             01:00:00
NOBELIUM            WEEKLY                SUNDAY             01:00:00
```

4.3.2.3 GROUP BY

The GROUP BY clause is used to summarize the output from a query by column name:

column_name rows to be grouped for resolving aggregate functions

This clause is generally used in conjunction with an aggregate function. The following example shows the sum of files and server storage used for each node:

```
adsm> select node_name, sum(num_files) as NUM_FILES, sum(logical_mb) as
       STORAGE_MB from occupancy group by node_name

NODE_NAME                NUM_FILES                               STORAGE_MB
------------------     -----------     --------------------------------
DANUBE                          20                                     0.70
NOBELIUM                        16                                    14.44
SEVERN                          25                                     2.30
```

4.3.2.4 HAVING

The HAVING clause is used to apply a condition to an aggregated column to limit the returned output:

predicate expression to filter output of aggregated values before displaying

The HAVING clause is used to include a condition for an aggregate function. The following example builds on the previous example by showing only those nodes that use greater than 1 MB of server storage:

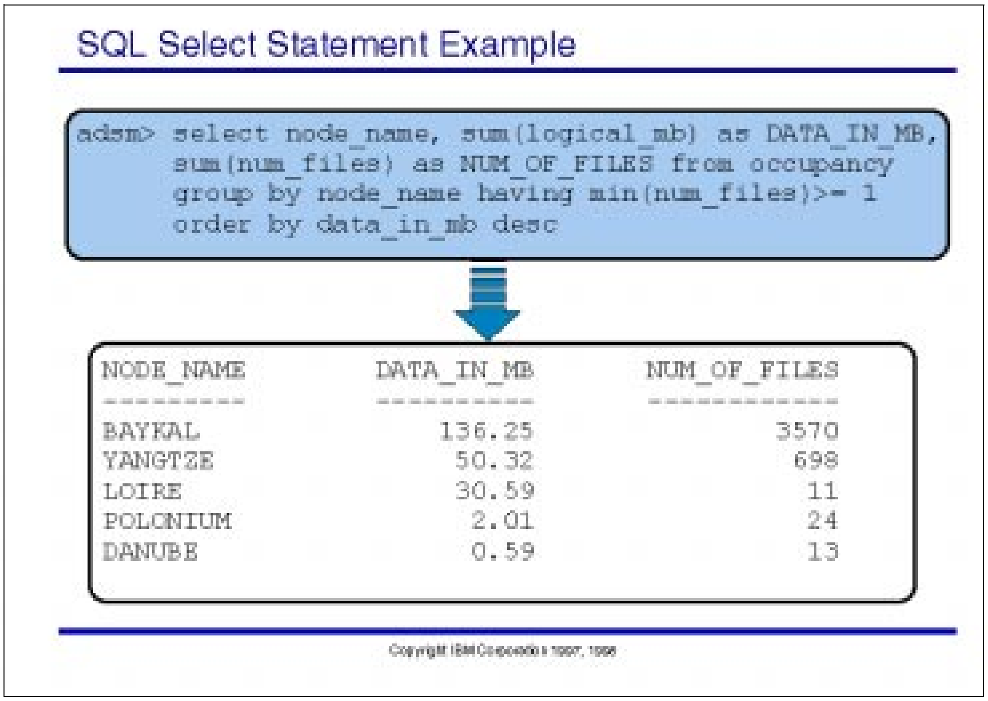
```
adsm> select node_name, sum(num_files) as NUM_FILES, sum(logical_mb)
       as STORAGE_MB from occupancy group by node_name having
        sum(logical_mb)>1

NODE_NAME                NUM_FILES                               STORAGE_MB
------------------     -----------     --------------------------------
NOBELIUM                        16                                    14.44
SEVERN                          25                                     2.30
```

4.3.2.5 ORDER BY

The ORDER BY clause is used to format the output returned from a query based on a sorted column:

output_column_name specification of output sorting, either in ascending (ASC) or descending (DESC) order; if more than one column_name is specified, the first column has primary sort order, then the second column, and so on.

“SQL Select Statement Example” on page 134. illustrates a sample SELECT statement using ORDER BY in conjunction with the other clauses discussed.



SQL Select Statement Example

```
adsm> select node_name, sum(logical_mb) as DATA_IN_MB,
      sum(num_files) as NUM_OF_FILES from occupancy
      group by node_name having min(num_files)>= 1
      order by data_in_mb desc
```

```
NODE_NAME            DATA_IN_MB          NUM_OF_FILES
---------            ----------          ------------
BAYKAL                   136.25                  3570
YANGTZE                   50.32                   698
LOIRE                     30.59                    11
POLONIUM                   2.01                    24
DANUBE                     0.59                    13
```

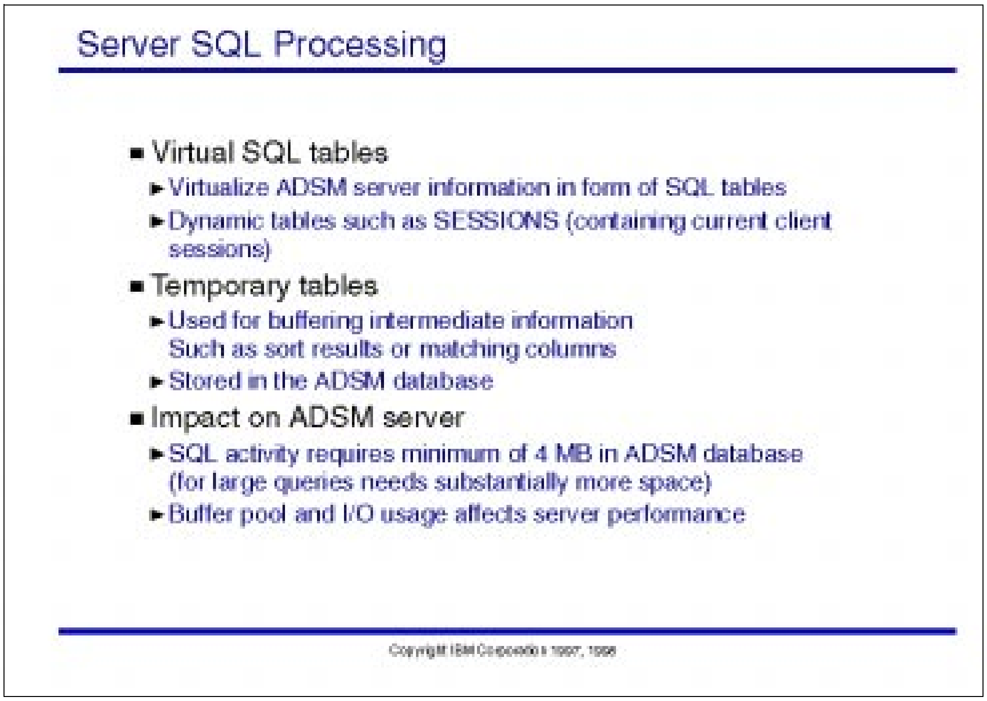
Copyright IBM Corporation 1997, 1998

4.3.3 SQL Select Statement Example

The graphic shows how a select statement can be used powerfully to manipulate the output report as desired.

An administrator may be interested in the amount of data that is used by each node. Information of this kind is stored in the occupancy table. Each row in the occupancy table holds information for one filespace of one node and of one type (for example, filespace is C drive, node is polonium, type is backup).

Instead of listing information for each filespace (through the QUERY OCCUPANCY command), the select statement groups and calculates information for each node. The aggregation function SUM is used to add up the number of files and the amount of data for each node. An aggregation function requires a GROUP BY clause that defines on which group of objects it has to operate. In the example, this is the node_name. The HAVING clause eliminates output lines of nodes that have not backed up at least one file. The output order is controlled by the ORDER BY clause. In this example it is ordered by the amount of data per node in descending order.



Server SQL Processing

- Virtual SQL tables
  - Virtualize ADSM server information in form of SQL tables
  - Dynamic tables such as SESSIONS (containing current client sessions)
- Temporary tables
  - Used for buffering intermediate information Such as sort results or matching columns
  - Stored in the ADSM database
- Impact on ADSM server
  - SQL activity requires minimum of 4 MB in ADSM database (for large queries needs substantially more space)
  - Buffer pool and I/O usage affects server performance

4.3.4 Server SQL Processing

SQL statements request output reports. To gather and format the requested information, the ADSM server uses virtual and temporary tables.

4.3.4.1 Virtual SQL Tables

ADSM server information is mapped into virtual tables to make it available for SQL select statements. This is also true for dynamic information. For example, the SESSIONS table can be queried with a select statement to display client sessions that are currently logged on to the server.

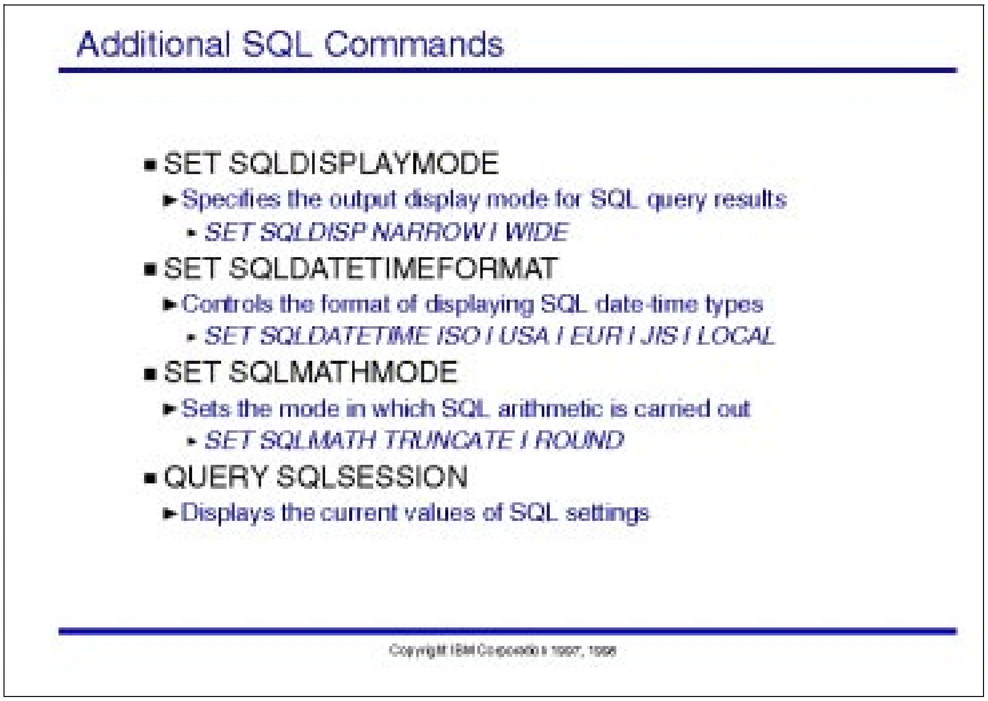
4.3.4.2 Temporary Tables

Many forms of SQL select statements generate intermediate data before the final select output report is displayed. Functions causing this are, for example, nested select statements, order conditions, or matching criteria. This intermediate data is stored in temporary tables. Temporary tables reside in the ADSM database in order to save memory resources. They are deleted on completion of the SELECT statement.

4.3.4.3 Impact on ADSM Server

Because it generates temporary tables, SQL processing requires at least a 4 MB partition in the ADSM database. For larger queries substantially more free space is needed.

Because SQL processing uses the database buffer pool and I/O resources, SELECT statements affect ADSM server performance. Very complicated or long-running select statements can slow down server performance significantly during processing.



Additional SQL Commands

- SET SQLDISPLAYMODE
  - Specifies the output display mode for SQL query results
    - *SET SQLDISP NARROW | WIDE*
- SET SQLDATETIMEFORMAT
  - Controls the format of displaying SQL date-time types
    - *SET SQLDATETIME ISO | USA | EUR | JIS | LOCAL*
- SET SQLMATHMODE
  - Sets the mode in which SQL arithmetic is carried out
    - *SET SQLMATH TRUNCATE | ROUND*
- QUERY SQLSESSION
  - Displays the current values of SQL settings

4.3.5 Additional SQL Commands

In addition to the SELECT command, other SQL-related commands are available for an administrator. These are used to control the output reports generated by the SELECT command.

4.3.5.1 SET SQLDISPLAYMODE

The SET SQLDISPLAYMODE command controls the manner in which SQL data types are displayed. It can be set to one of the following values:

Narrow Column display width is 18; any wider string is displayed on multiple lines

Wide Column display width is 250

4.3.5.2 SET SQLDATETIMEFORMAT

The SET SQLDATETIMEFORMAT command controls the format in which SQL date and time data are displayed. One of the following formats can be chosen:

ISO	International Standard Organization
USA	IBM USA standard
EUR	IBM European standard
JIS	Japanese industrial standard
Local	Site-defined

Currently JIS and Local are the same as ISO.

4.3.5.3 SET SQLMATHMODE

The SET SQLMATHMODE command defines the mode in which SQL arithmetic is carried out. There are two options:

Truncate Decimal numbers are truncated

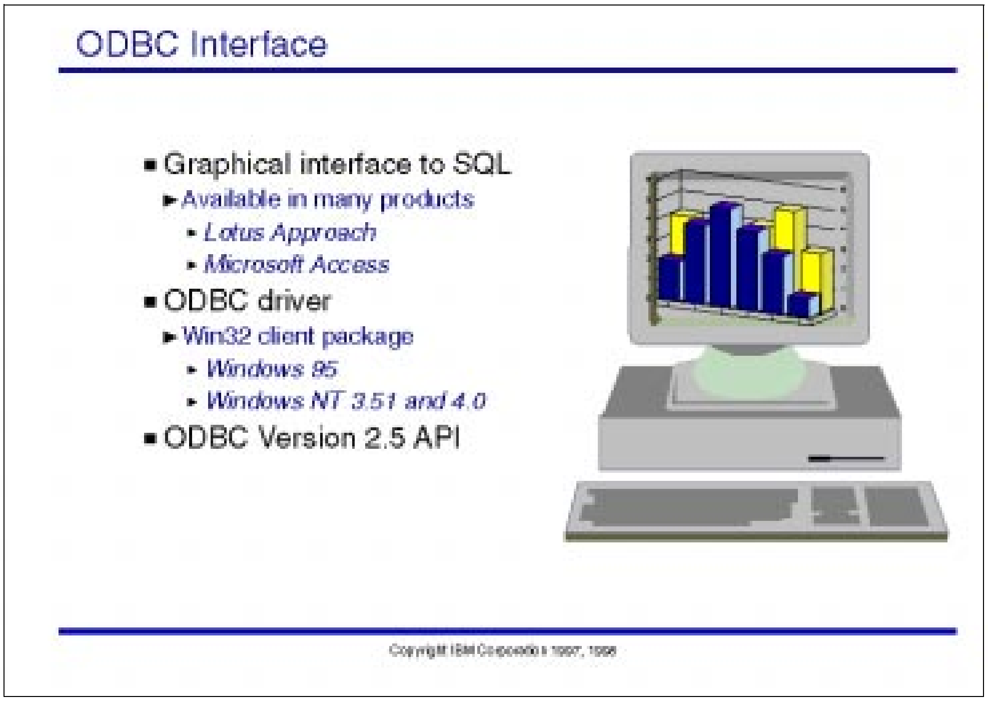
Round Decimal numbers are rounded

4.3.5.4 QUERY SQLSESSION

The QUERY SQLSESSION command displays the current SQL settings:

```
adsm> query sqlsession

Column      Date-Time     Arithmetic     Cursors
Display     Format        Mode           Allowed?
Format
--------    ---------     ----------     --------
Narrow         ISO         Truncate        Yes
```

The “Cursors Allowed?” column indicates the SQL cursor support used for ODBC.



ODBC Interface

- Graphical interface to SQL
  - Available in many products
    - *Lotus Approach*
    - *Microsoft Access*
- ODBC driver
  - Win32 client package
    - *Windows 95*
    - *Windows NT 3.51 and 4.0*
- ODBC Version 2.5 API

Copyright IBM Corporation 1997, 1998

4.3.6 ODBC Interface

In ADSM Version 3, to complement the ability to use SQL to query the contents of the ADSM database, an ODBC interface has also been implemented. This interface enables the output from the ADSM server database to be manipulated by other ODBC compliant applications.

4.3.6.1 Graphical SQL Interface

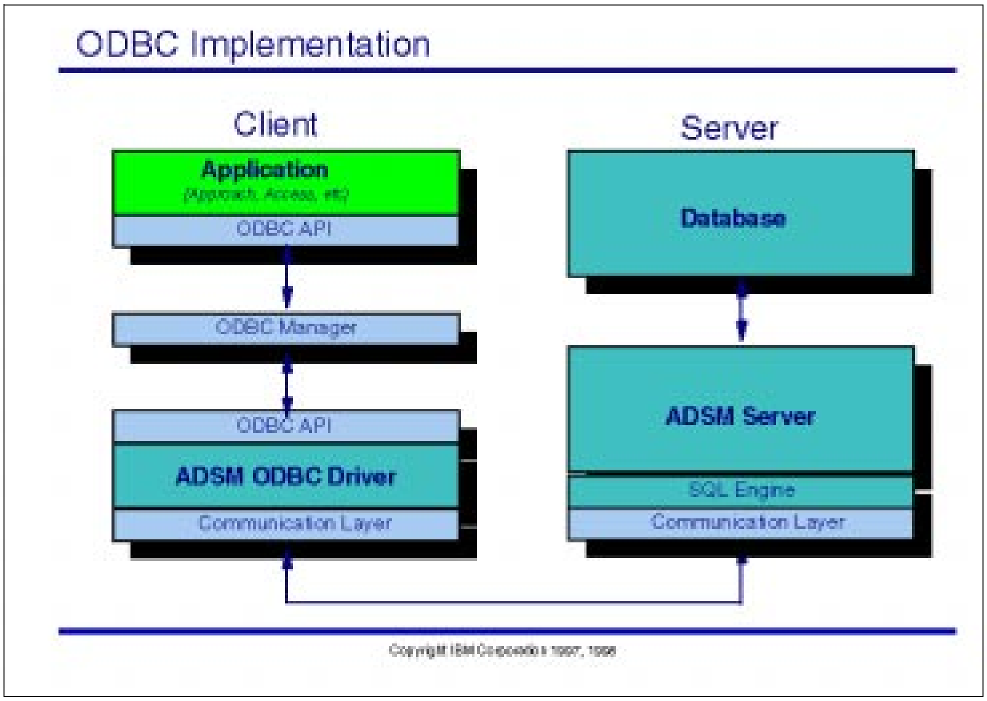
ODBC is a standard interface between database engines and front ends. It enables products such as Lotus Approach or Microsoft Access to be used to graphically construct SQL select statements, which are then dispatched to the database. The information produced by these select statements is then returned in tabular form, enabling further processing, usually the production of graphs or tables.

4.3.6.2 ODBC Driver

The ODBC driver supplied with Version 3 is part of the client package. The ODBC client application, Lotus Approach, for example, interfaces with the ADSM client, which in turn interacts with the ADSM server. This function is available only for the following Win32 clients: Windows 95, Windows NT 3.51, and Windows 4.0.

4.3.6.3 ODBC Version 2.5 API

The ODBC driver in Version 3 supports the ODBC 2.5 API. The ADSM SQL interface supports read-only SQL SELECT statements and does not allow any alteration of the information in the database. Thus the interface does not completely conform to any of the ODBC API or SQL grammar conformance levels. Applications that require particular conformance levels may have problems connecting to ADSM.



ODBC Implementation

Client

Application

(Approach, Access, etc)

ODBC API

ODBC Manager

ODBC API

ADSM ODBC Driver

Communication Layer

Server

Database

ADSM Server

SQL Engine

Communication Layer

Copyright IBM Corporation 1997, 1998

4.3.7 ODBC Implementation

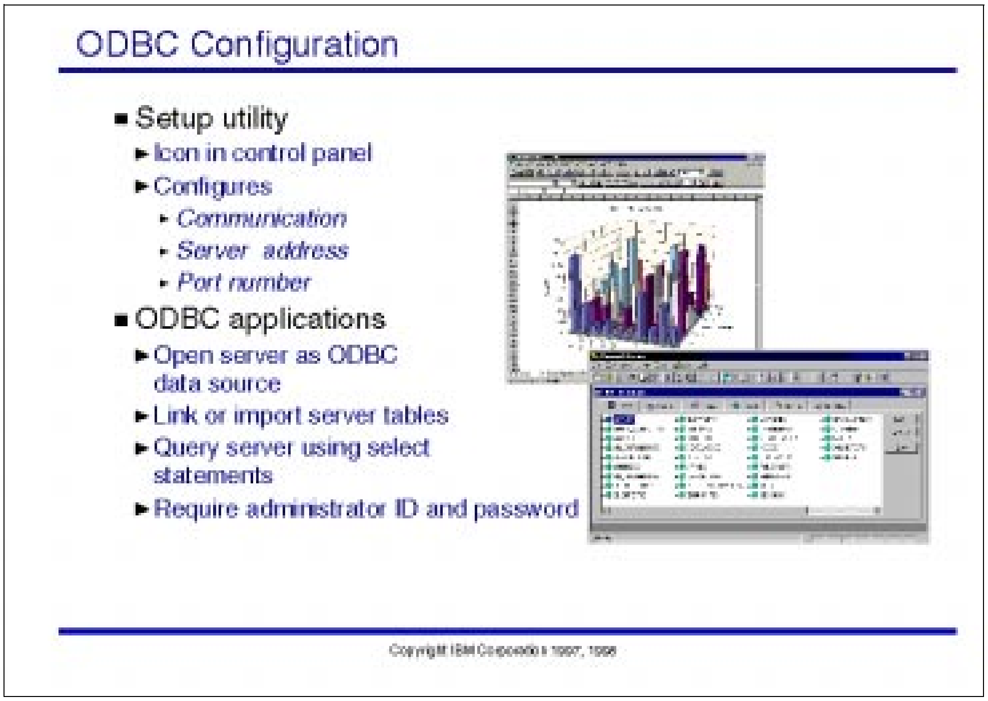
The graphic illustrates the flow of information, both to and from the server, during the processing of an SQL SELECT operation from an ODBC application.

4.3.7.1 Client

The ODBC interface through which the client application accesses the server is built into the ADSM administrative client. Using the ODBC API that interfaces to the ADSM ODBC driver, the client application submits requests to the common code of the ADSM client. At this point a login prompt is displayed in the client application, and a valid administrative ID and password must be provided. An administrative session is then started with the ADSM server through normal client communications.

4.3.7.2 Server

The server, on receiving the ODBC request, starts an administrative session, which may be tracked with the QUERY SESSION command. The administrative session then submits the SQL SELECT statement to the server, which in turn queries the database for the information. Once the results of the query have been generated, they are passed back through the common client code to the ODBC API in the application, and the session is closed.



ODBC Configuration

- Setup utility
  - Icon in control panel
  - Configures
    - *Communication*
    - *Server address*
    - *Port number*
- ODBC applications
  - Open server as ODBC data source
  - Link or import server tables
  - Query server using select statements
  - Require administrator ID and password

Copyright IBM Corporation 1997, 1998

4.3.8 ODBC Configuration

To ensure that the ODBC implementation is as flexible as possible, ODBC connectivity is implemented through a series of ODBC driver within the operating system. Thus the operating system can offer ODBC connectivity in a standardized way to all ODBC-compliant applications.

4.3.8.1 Setup Utility

The Version 3 client package includes the ODBC driver and a copy of the ODBC Database Source Administrator. At client installation the ODBC driver is an optional component. If selected, an ODBC icon is created in the control panel. This icon starts the ODBC Data Source Administrator, which can be used to configure the ADSM driver as a user, system or file data source. User data sources are available only to the defining user. System data sources can be accessed by any user of the system. To configure the ADSM ODBC driver as a data source, the server name, TCP/IP address, and TCP port number must be provided.

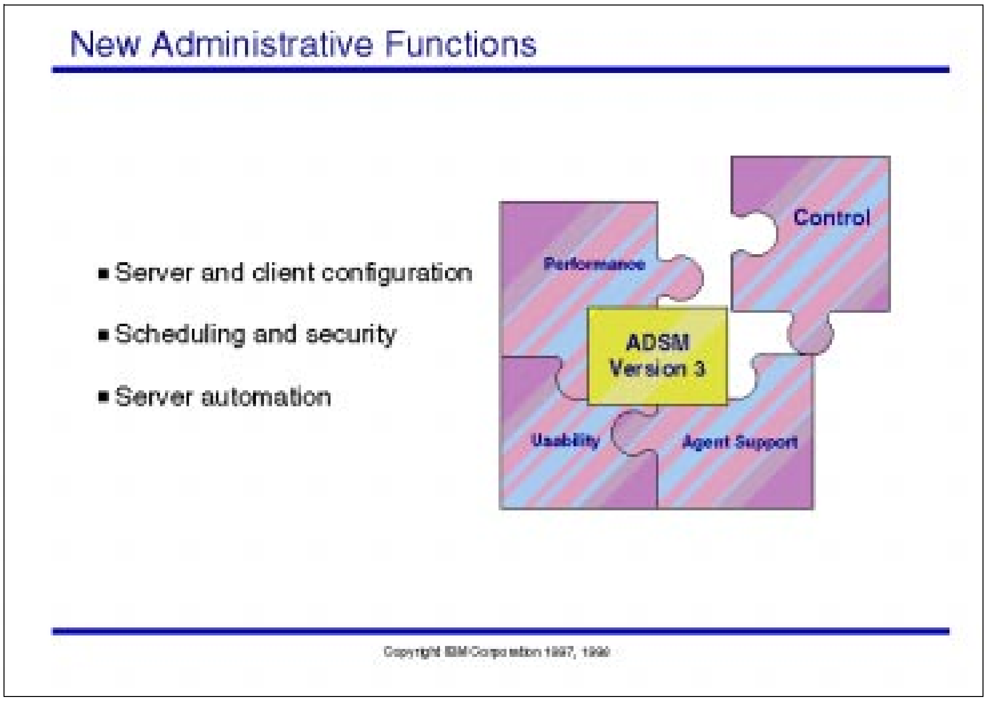
4.3.8.2 ODBC Applications

An application running in an environment with the ADSM ODBC driver installed and configured as a data source can import or link to ADSM database tables. This link is normally done through an Open ODBC Data Source or Get External Data menu item. It is important to note that import imports all data associated with the tables as well as the tables themselves. Standard ODBC supports database modification as well as query. However, the ADSM database is available only as read only, and an ODBC application cannot update it.

Once local application tables are linked to their equivalent server tables, the application can then construct SQL queries. The results can be manipulated in the application or saved and used with a third application such as a graphics package or report generator.

Server authentication for ODBC applications is performed in the same way as for other administrative interfaces. When the application accesses the server, a prompt is issued for an administrator user ID and password. These must be entered before the application can access the server.

Chapter 5. New Administrative Functions



New Administrative Functions

- Server and client configuration
- Scheduling and security
- Server automation

Copyright IBM Corporation 1997, 1998

This chapter introduces various administration enhancements that have been made in Version 3. The following topics are covered:

- Server and client configuration

  This section covers enhancements to server configuration, client option sets, and enhanced client information.

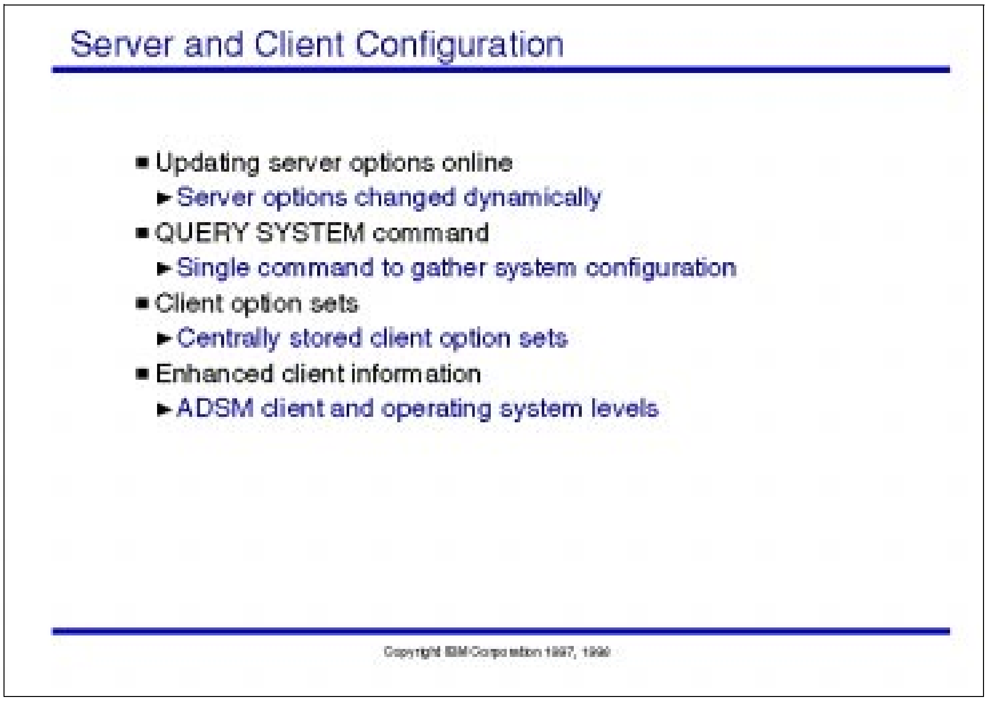
- Scheduling and security

  This section covers new client scheduling options, client signon, and password security.

- Server automation

  This section covers server scripts, database or log space expansion, and synchronous administrative commands.

5.1 Server and Client Configuration



Server and Client Configuration

- Updating server options online
  - Server options changed dynamically
- QUERY SYSTEM command
  - Single command to gather system configuration
- Client option sets
  - Centrally stored client option sets
- Enhanced client information
  - ADSM client and operating system levels

This section introduces the enhancements that have been made to Version 3 in terms of server and client configuration:

- Updating server options online

  In Version 3 new functions have been added to enable you to change certain server options online so that you do not have to stop and restart the ADSM server to change options.

- QUERY SYSTEM Command

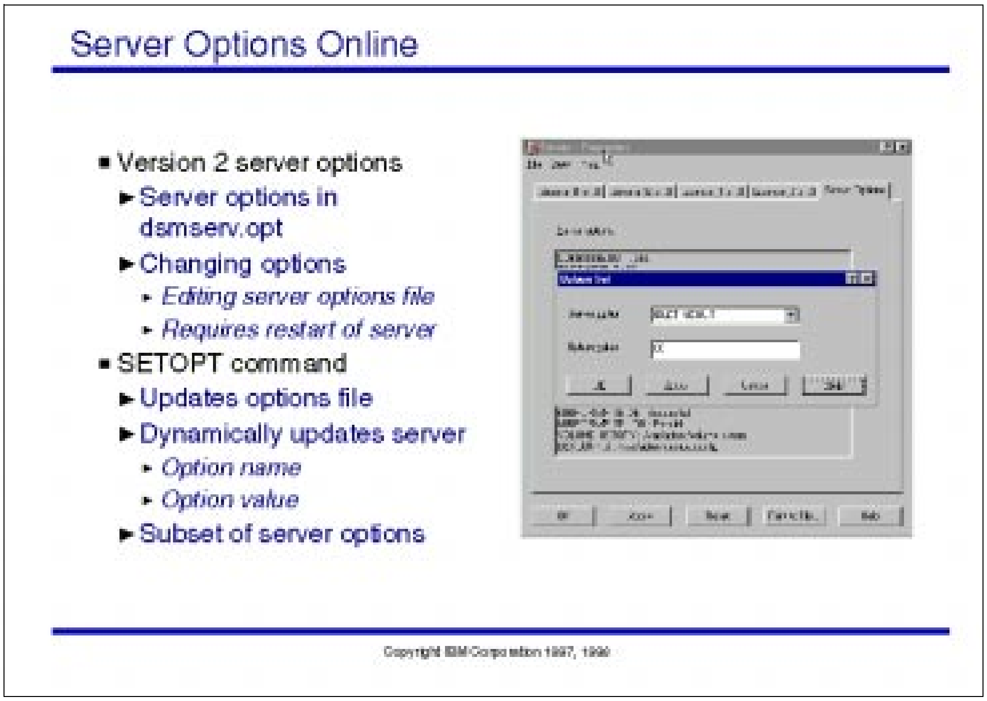
  The new QUERY SYSTEM command gathers information that provides a high-level overview of the ADSM server configuration and capacity.

- Client option sets

  Version 3 introduces the concept of client option sets. These are groups of client options, defined at the server, that can then be allocated to client nodes.

- Enhanced client information

  To make the management of client nodes within the enterprise less complicated, the information that a client passes to the server at successful login has been enhanced.



Server Options Online

- Version 2 server options
  - Server options in dsmserv.opt
  - Changing options
    - *Editing server options file*
    - *Requires restart of server*
- SETOPT command
  - Updates options file
  - Dynamically updates server
    - *Option name*
    - *Option value*
  - Subset of server options

Copyright IBM Corporation 1997, 1998

5.1.1 Server Options Online

To enable ADSM to be flexible while maintaining normal levels of service, the server has been updated to enable options to be changed while the server remains online.

5.1.1.1 Version 2 Server Options

In Version 2, almost all server options are defined in the dsmserv.opt file. This file is read by the ADSM server only on startup, and any changes made to the file are picked up only at that time. Therefore, to change server options, you have to stop and restart the ADSM server.

5.1.1.2 SETOPT Command

In Version 3 a new command, SETOPT, has been added to change certain server options online. Thus server options can be changed without restarting the server to force the dsmserv.opt file to be reread.

To change server options use the SETOPT command. This command appends changed options to the end of the dsmserv.opt file and dynamically changes the option on the server:

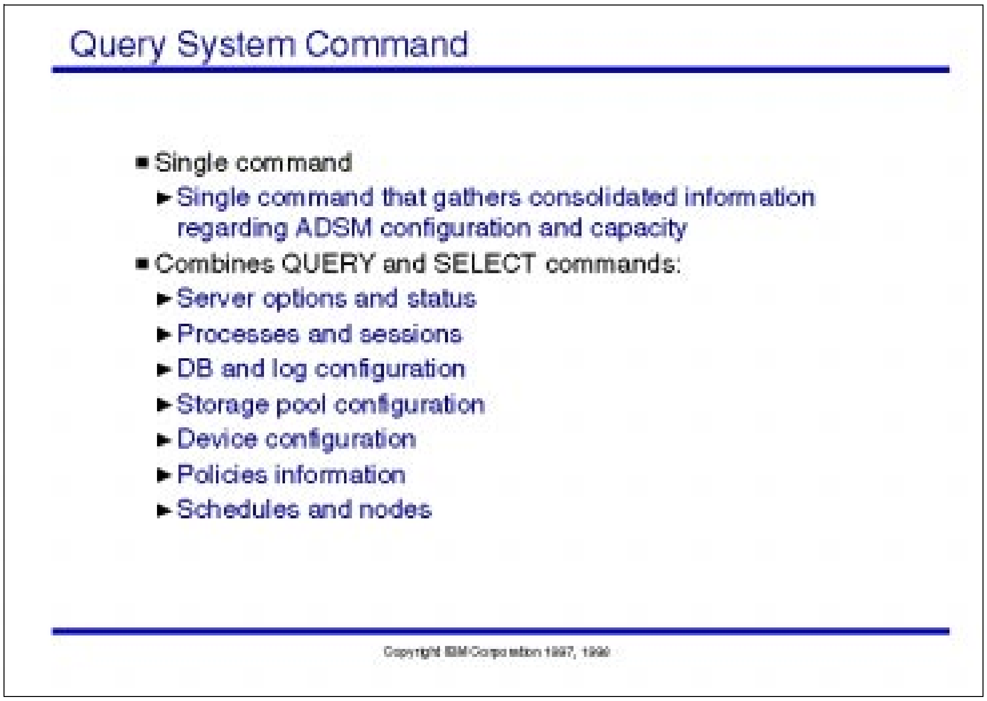
```
adsm> setopt dateformat 5

Do you wish to proceed? y

ANR2119I The DATEFORMAT option has been changed in the options file
```

Any options changed using the SETOPT command are actioned immediately on the server.

The command requires an option name and option value. The option value has a 255-character limit and should be provided in quotes if it involves embedded blanks.

The SETOPT command supports only a subset of server options:

- COMMTIMEOUT
- IDLETIMEOUT
- DATEFORMAT
- TIMEFORMAT
- NUMBERFORMAT
- MAXSESSIONS
- EXPINTERVAL
- EXPQUIET
- NOAUDITSTORAGE



Query System Command

- Single command
  - Single command that gathers consolidated information regarding ADSM configuration and capacity
- Combines QUERY and SELECT commands:
  - Server options and status
  - Processes and sessions
  - DB and log configuration
  - Storage pool configuration
  - Device configuration
  - Policies information
  - Schedules and nodes

5.1.2 Query System Command

In many situations it is helpful to both the administrator and other support personnel to obtain details of the server configuration. In Version 2, this could be done only by creating a macro or capturing the output from a large number of single commands. In Version 3 the QUERY SYSTEM command provides this detailed server configuration information.

5.1.2.1 Single Command

The QUERY SYSTEM command allows the administrator to issue a single command and be provided with detailed server configuration and capacity information.

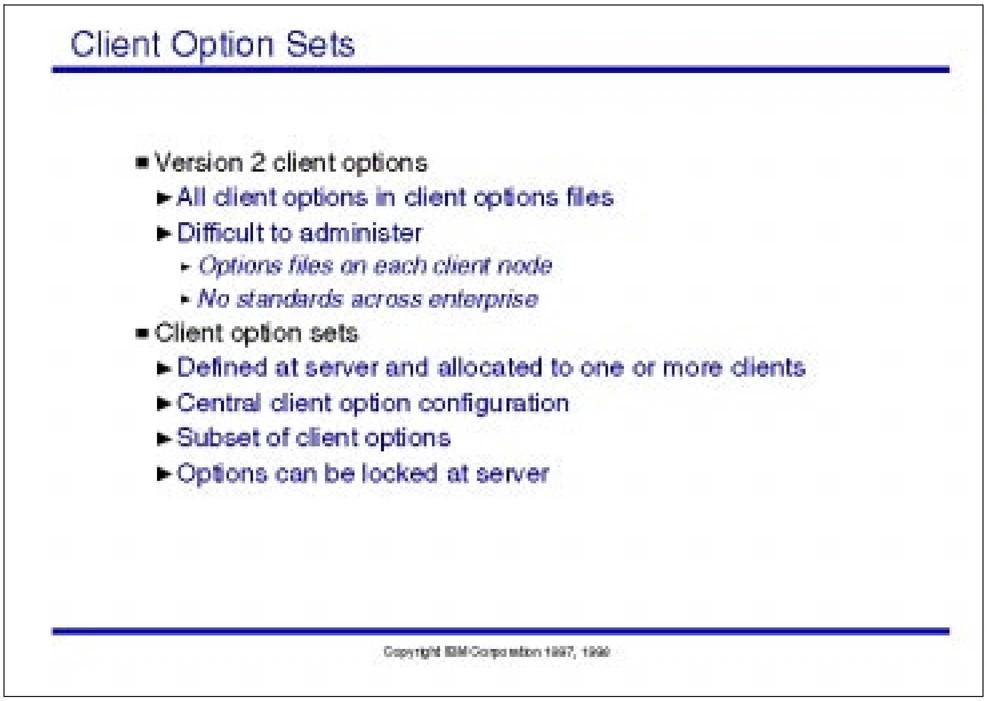
5.1.2.2 Combines QUERY and SELECT Commands

The new command consolidates the output from a number of QUERY commands along with a number of new SQL SELECT commands. With this approach, the query commands can be used to obtain detailed information about specific areas of the server, and the SELECT statements can be used to produce summaries of certain key areas.

The information included in the output from a QUERY SYSTEM command is a formatted version of the output from the following commands:

- QUERY STATUS
- QUERY OPTIONS
- QUERY PROCESS
- QUERY SESSION
- QUERY DB F=D

- QUERY DBVOLUME * F=D
- QUERY LOG F=D
- Q LOGVOLUME * F=D
- Q VOLUME
- QUERY STGPOOL * F=D
- QUERY DOMAIN
- QUERY MGMTCLASS
- QUERY COPYGROUP * * * STANDARD TYPE=BACKUP
- QUERY COPYGROUP * * * STANDARD TYPE=ARCHIVE
- QUERY SCHEDULE
- QUERY ASSOCIATION
- QUERY DEVCLASS * F=D
- select platform_name, count(*) from nodes group by platform_name
- select stgpool_name,devclass_name,count(*) from volumes group by stgpool_name,devclass_name



Client Option Sets

- Version 2 client options
  - All client options in client options files
  - Difficult to administer
    - *Options files on each client node*
    - *No standards across enterprise*
- Client option sets
  - Defined at server and allocated to one or more clients
  - Central client option configuration
  - Subset of client options
  - Options can be locked at server

Copyright IBM Corporation 1997, 1998

5.1.3 Client Option Sets

One of the challenges facing ADSM administrators is the ever-increasing number of clients managed by a server. In such an environment each client requires configuration through a local options file. Management of such an environment becomes more and more complex as the number of clients and the number of configuration options increase.

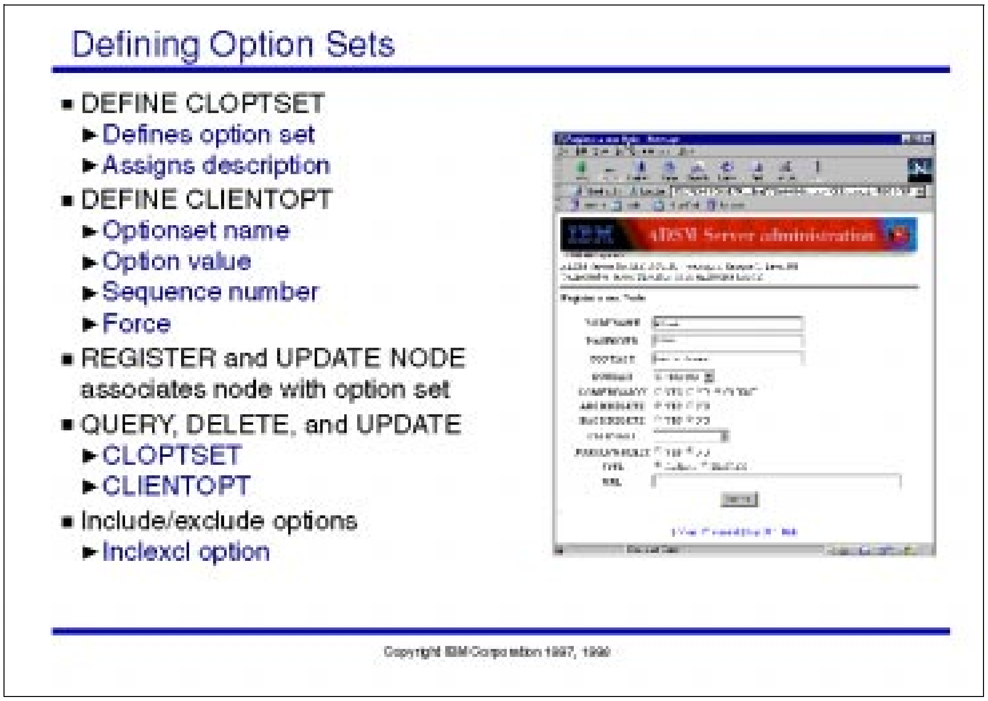
5.1.3.1 Version 2 Client Options

In Version 2, all client options are defined in one or two client options files, dsm.sys (Unix clients only) and dsm.opt. Therefore, in a distributed environment direct access to the client machine is required to edit these files. Global option changes are thus time consuming and complicated, and the user of each machine can change the options at will. For most options, a central policy cannot be defined or enforced without a great deal of difficulty.

5.1.3.2 Client Option Sets

Version 3 introduces the client option set, a set of client options defined at the server that can be allocated to one or more clients either at registration or during normal operation. The option set can then be updated once to change all clients defined to it. Thus a number of client options can be configured centrally and served to clients as required.

The options defined in a client option set are a subset of the available client options. Options such as communications are still stored on the client machine. To allow the administrator to control these options centrally, it is also possible to specify that individual options cannot be overridden in the client’s local option file. Thus one set of default values can be defined for each type of client in the environment, and the client machines can still be customized, within acceptable limits.



Defining Option Sets

- DEFINE CLOPTSET
  - Defines option set
  - Assigns description
- DEFINE CLIENTOPT
  - Optionset name
  - Option value
  - Sequence number
  - Force
- REGISTER and UPDATE NODE associates node with option set
- QUERY, DELETE, and UPDATE
  - CLOPTSET
  - CLIENTOPT
- Include/exclude options
  - Inclexcl option

IBM
ADSM Server administration

Copyright IBM Corporation 1997, 1998

5.1.4 Defining Option Sets

Defining a client option set is a two stage process: (1) the client option set is defined, and (2) the options are defined to it.

5.1.4.1 DEFINE CLOPTSET

The DEFINE CLOPTSET command defines the option set name and description. The name can be up to 64 characters long and should not include embedded blanks. The option set description can be up to 255 characters long and should be enclosed in quotes if embedded blanks are used. To issue this command, an administrator needs to have SYSTEM or unrestricted POLICY privilege.

5.1.4.2 DEFINE CLIENTOPT

Once the option set has been defined to ADSM, options and their values must be defined to the option set through the DEFINE CLIENTOPT command. The option set name, option name, and option value must be supplied. Optionally a sequence number and/or force value can be supplied.

The sequence number is assigned to a client option if the same option is defined more than once. Options such as include can be defined to a client option set many times, and their sequence is important. Default sequence numbers are applied in the order in which the options are defined. You can alter this sequence by use of the sequence number.

The FORCE option is used to specify options that cannot be overridden by the client’s local option file. To stop an option from being overridden by the client, define it to the option set with FORCE=YES. The default is NO.

5.1.4.3 REGISTER and UPDATE NODE

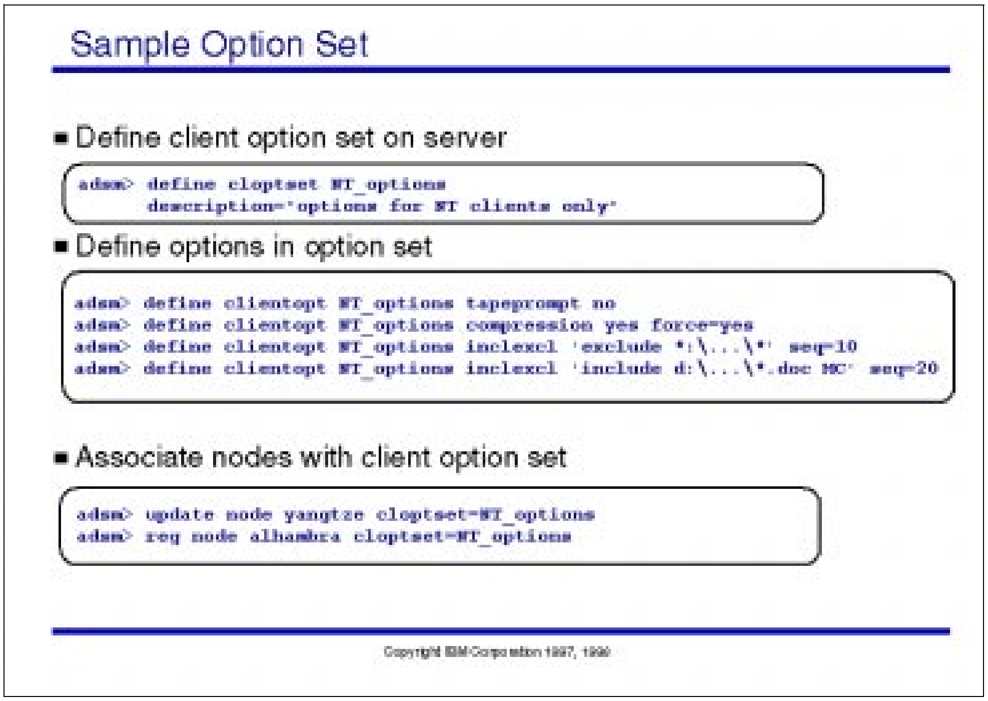
Once a client option set and the client options have been defined, nodes can be associated with that set through either the CLOPTSET option on the REGISTER NODE command, for new nodes, or on the UPDATE NODE command, for existing nodes.

5.1.4.4 QUERY, DELETE, and UPDATE

Client option sets can be queried, deleted, or updated through the new QUERY, DELETE, and UPDATE CLOPTSET commands. Individual client options within an option set can be queried or deleted through the new QUERY and DELETE CLIENTOPT commands. The sequence number of a client option can be changed with the UPDATE CLIENTOPT command.

5.1.4.5 Include/Exclude Option

The INCLEXCL option, also used on UNIX clients to define a file containing include and exclude statements, can be used in a client option set to define a sequence of include and exclude statements. This option is used to define each line in a sequence of include or exclude statements. The details of the file to be processed, or wildcard statements, are enclosed in quotes along with any management class that should be bound to files being backed up.



Sample Option Set

- Define client option set on server

```
adsm> define cloptset NT_options
      description='options for NT clients only'
```

- Define options in option set

```
adsm> define clientopt NT_options tapeprompt no
adsm> define clientopt NT_options compression yes force=yes
adsm> define clientopt NT_options inclexcl 'exclude *:\...\*' seq=10
adsm> define clientopt NT_options inclexcl 'include d:\...\*.doc MC' seq=20
```

- Associate nodes with client option set

```
adsm> update node yangtze cloptset=NT_options
adsm> reg node alhambra cloptset=NT_options
```

5.1.5 Sample Option Set

The graphic shows the steps taken to define a client option set, add client options to the option set, and associate nodes with the client option set.

5.1.5.1 Define Client Option Set on Server

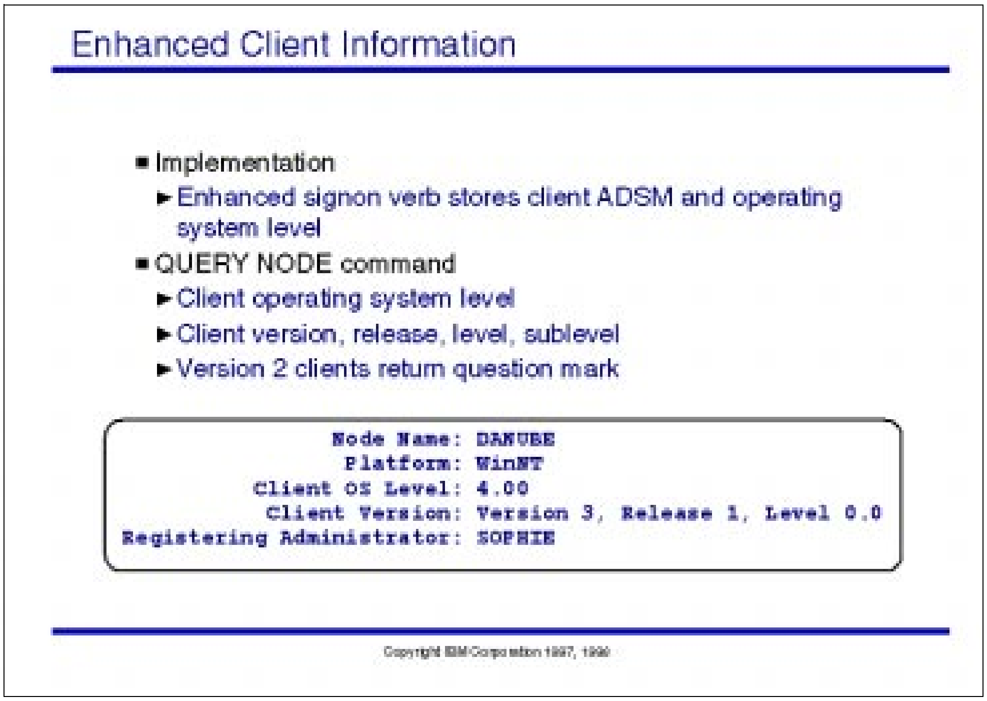
The first stage is to define the option set to ADSM. The command shows a client option set being defined with the NT_options name and an "options for NT clients only" description.

5.1.5.2 Define Options in Option Set

Once the option set has been defined, the DEFINE CLIENTOPT command is used to specify options within that option set. In the example clients are not allowed to override the COMPRESSION option. A global exclude statement for all files is followed by an include statement for any file with an extension of .doc. Files backed up with this include statement are bound to the MC management class. The seq numbers 10 and 20 are used to ensure that these two inclexcl statements are processed in the correct sequence.

5.1.5.3 Associate Nodes with Client Option Set

The final stage is to associate client nodes with the option set. As shown, you make the association, using either the REGISTER NODE or UPDATE NODE command.



Enhanced Client Information

- Implementation
  - Enhanced signon verb stores client ADSM and operating system level
- QUERY NODE command
  - Client operating system level
  - Client version, release, level, sublevel
  - Version 2 clients return question mark

```
              Node Name: DANUBE
               Platform: WinNT
        Client OS Level: 4.00
         Client Version: Version 3, Release 1, Level 0.0
Registering Administrator: SOPHIE
```

Copyright IBM Corporation 1997, 1998

5.1.6 Enhanced Client Information

When a node connects to the server in ADSM, a signon verb is sent to inform the server of certain information about the client. In Version 2 this information consists of only the client platform name. This information is determined at the first client connection and never changed.

5.1.6.1 Implementation

In Version 3 this information has been enhanced to include the client operating system and ADSM client level. Thus administrators can obtain more detailed information about the nodes in the environment. This information is updated at the server each time the client connects successfully.

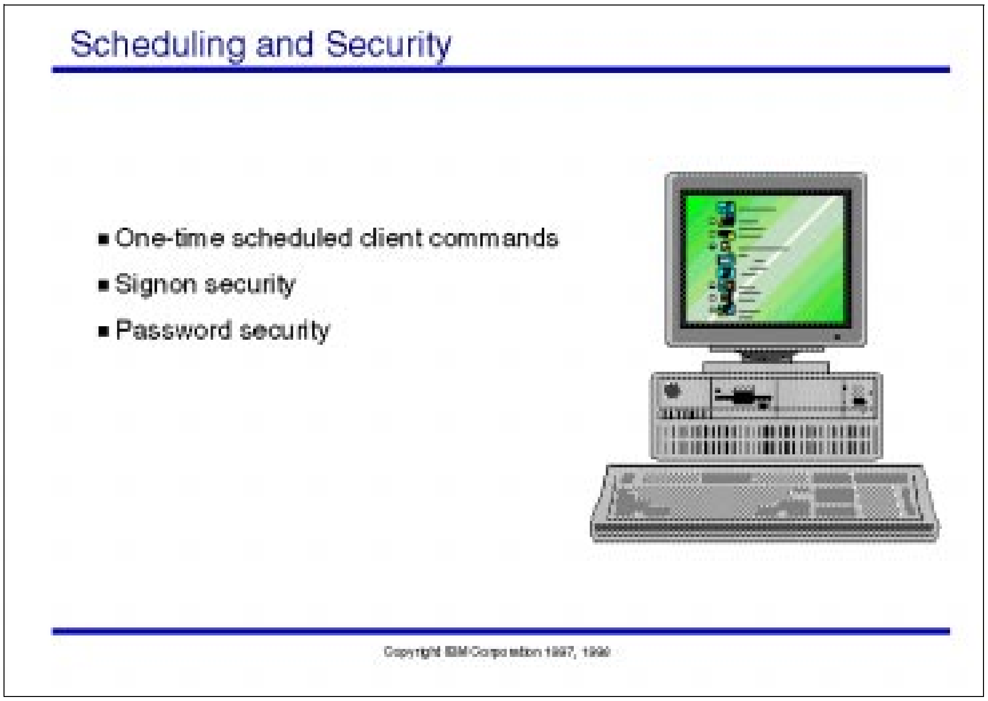
5.1.6.2 QUERY NODE Command

The administrator can, by use of either the QUERY NODE FORMAT=DETAILED or SELECT command querying the NODES table, gather information and reports about all nodes in the environment. This information can then be used to track changes in ADSM client code level or operating system level of every node. The new signon verb returns the following information:

- Client operating system level
- Client version
- Client release
- Client level
- Client sublevel

As Version 2 clients do not provide this enhanced signon information, the output of QUERY NODE contains question marks in the new fields.

5.2 Scheduling and Security



Scheduling and Security

- One-time scheduled client commands
- Signon security
- Password security

Copyright IBM Corporation 1997, 1998

In addition to the administrative enhancements for new functions and interfaces, there are a number of miscellaneous enhancements to assist the ADSM administrator:

- One-time scheduled client commands

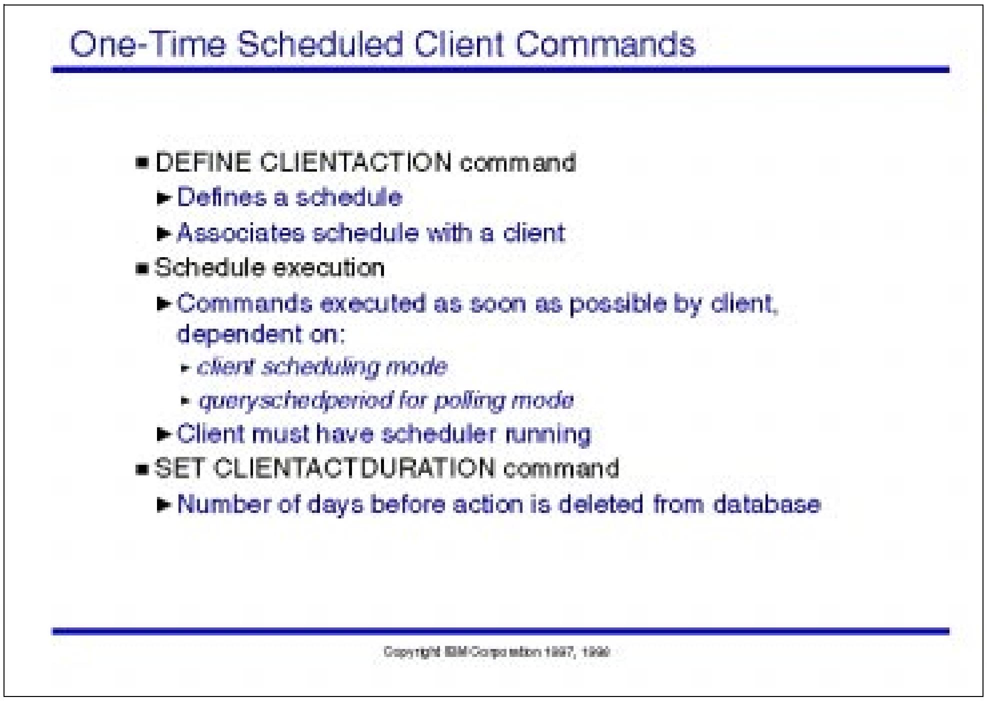
  Reduce the work required by an administrator when issuing a one-time schedule to a client or number of clients by providing fast-actioned, automatically deleted client action schedules

- Signon security

  Provides database tracking of node and administrator invalid logins and centralized control of lockout

- Password security

  New administrator functions centralize control of password restrictions for ADSM administrators and nodes.



One-Time Scheduled Client Commands

- DEFINE CLIENTACTION command
  - Defines a schedule
  - Associates schedule with a client
- Schedule execution
  - Commands executed as soon as possible by client, dependent on:
    - *client scheduling mode*
    - *queryschedperiod for polling mode*
  - Client must have scheduler running
- SET CLIENTACTDURATION command
  - Number of days before action is deleted from database

Copyright IBM Corporation 1997, 1998

5.2.1 One-Time Scheduled Client Commands

In Version 2 client schedules have to be defined in advance and then executed at the specified time. If the schedule is for a single operation, the schedule should be deleted from the server after completion.

5.2.1.1 DEFINE CLIENTACTION Command

A new command has been introduced to enable administrators to execute one-time client scheduled commands. DEFINE CLIENTACTION enables an administrator to have a command executed on a client without defining a schedule and associating the schedule with the client. With this command the command executes once only.

Clients can be selected from multiple domains through the NODELIST and DOMAIN parameters. Pattern-matching expressions can be used in the NODELIST specified; if no nodes are selected, the action is sent to all nodes.

5.2.1.2 Schedule Execution

When a command is specified with the DEFINE CLIENTACTION command, the speed with which that command is executed depends entirely on the scheduling mode for the desired client. If the client schedule mode is server-prompted, the command is executed immediately. If the client is using a schedule mode of POLLING, the action is executed the next time the client queries the server for its schedules (governed by the QUERYSCHEDPERIOD client parameter).

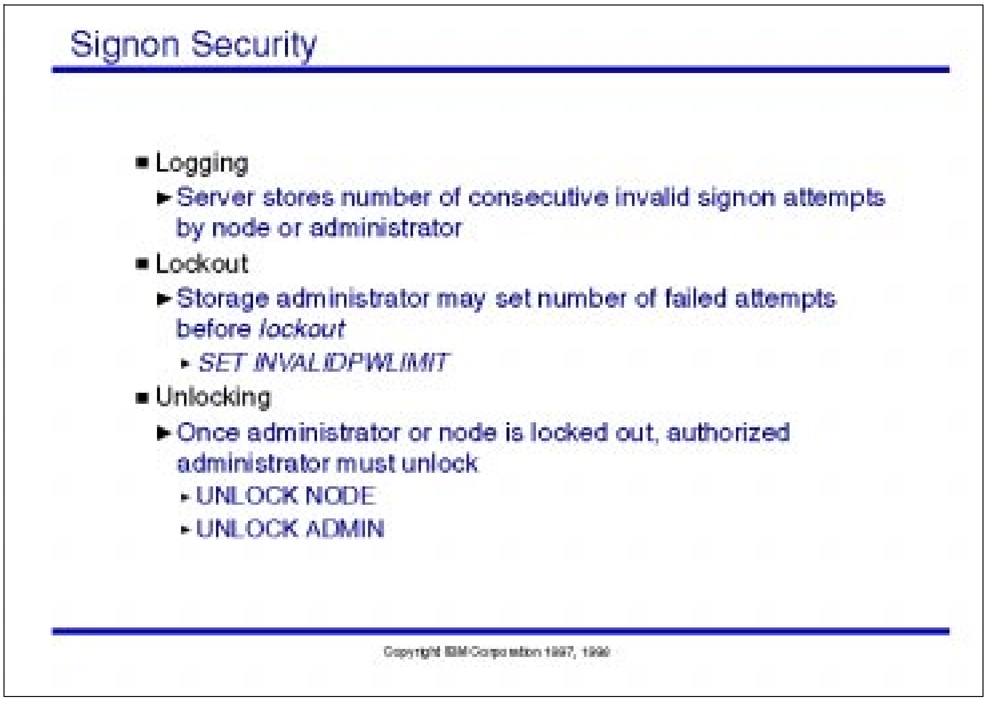
Clients for which actions are defined must have their scheduled client running.

5.2.1.3 SET CLIENTACTIONDURATION Command

The new SET CLIENTACTDURATION command specifies the duration in days for the schedule and the number of days it is to be retained in the ADSM database before it is automatically deleted. If this value is set to 0, the schedule duration is indefinite and the schedule and associations are not to be deleted. Note that only schedules and associations defined by the DEFINE CLIENTACTION are deleted from the ADSM database automatically.

The value of CLIENTACTDURATION can be viewed through the QUERY STATUS command. The defaults for a client action are:

- Schedule priority is set to 1
- Duration is set to 5 days
- Period is set to Onetime



Signon Security

- Logging
  - Server stores number of consecutive invalid signon attempts by node or administrator
- Lockout
  - Storage administrator may set number of failed attempts before *lockout*
    - *SET INVALIDPWLIMIT*
- Unlocking
  - Once administrator or node is locked out, authorized administrator must unlock
    - UNLOCK NODE
    - UNLOCK ADMIN

5.2.2 Signon Security

To enhance the security of ADSM, new methods of detecting unauthorized attempts to access the server by either administrative or backup-archive clients have been introduced in Version 3.

In Version 2, although any unauthorized attempts to connect were logged on the activity log, nowhere in the server was a record kept of the number of consecutive unsuccessful login attempts. Therefore if it was suspected that such attempts were being made, tracking was only possible by either manually checking the activity log or by writing scripts to search for the failed attempts and report accordingly.

5.2.2.1 Logging

In Version 3, server security has been enhanced in that ADSM now keeps a record of the number of consecutive unsuccessful login attempts by any node or administrator. This counter increments each time a client or administrator fails signon authentication, and it is reset to 0 when signon completes successfully. The current value of the counter for each node or administrator can be viewed through the QUERY NODE F=D or QUERY ADMIN F=D commands. Alternatively, a SELECT statement can be used of this form:

```
adsm> select node_name,invalid_pw_count from nodes

      NODE_NAME              INVALID_PW_COUNT
      ------------------     ----------------
      BAYKAL                                1
      CLIENT                                0
      DANUBE                                0
      FERMIUM                               0
      LOIRE                                 2
      MALMO                                 0
      POLONIUM                              0
```

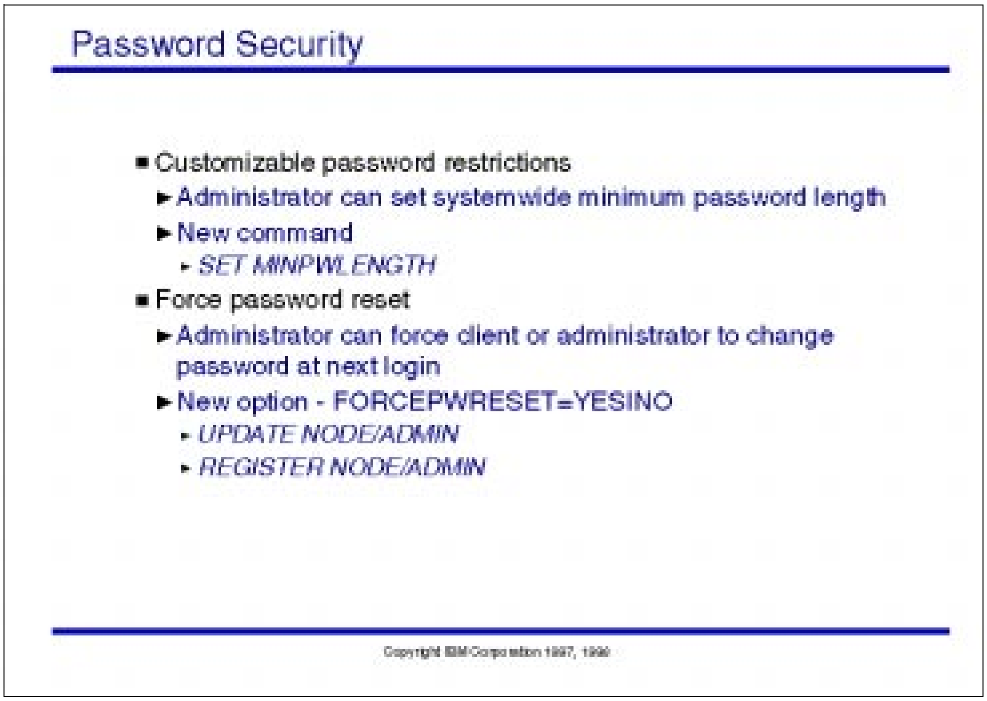
5.2.2.2 Lockout

The ADSM administrator can now specify the number of consecutive invalid attempts a node or administrator may have before he or she is locked out of the ADSM server. The SET INVALIDPWLIMIT command is used to specify the limit, and the QUERY STATUS command displays the current value. This is a global value for the server. The exception is that ADSM does not lock out the last administrator in the system, even if the limit is exceeded.

5.2.2.3 Unlocking

Once an administrator or node has been locked out of the ADSM server, another administrator with SYSTEM privilege must use either the UNLOCK NODE or UNLOCK ADMIN command to unlock them. The ways in which the number of failed logins counter may be reset to zero is:

- Node or administrative client (that is not locked) logs on successfully
- Node or administrative client (that is not locked) resets its password
- Administrator sets or resets the client's password
- Administrator can unlock, using the UNLOCK NODE or UNLOCK ADMIN command.



Password Security

- Customizable password restrictions
  - Administrator can set systemwide minimum password length
  - New command
    - *SET MINPWLENGTH*
- Force password reset
  - Administrator can force client or administrator to change password at next login
  - New option - FORCEPWRESET=YES|NO
    - *UPDATE NODE/ADMIN*
    - *REGISTER NODE/ADMIN*

Copyright IBM Corporation 1997, 1999

5.2.3 Password Security

In Version 2 the password selected for an administrator or a node is not controlled by any serverwide template. Thus nodes or administrators can select their own passwords with any length. There is no way for an administrator to force nodes or administrators to change their own passwords.

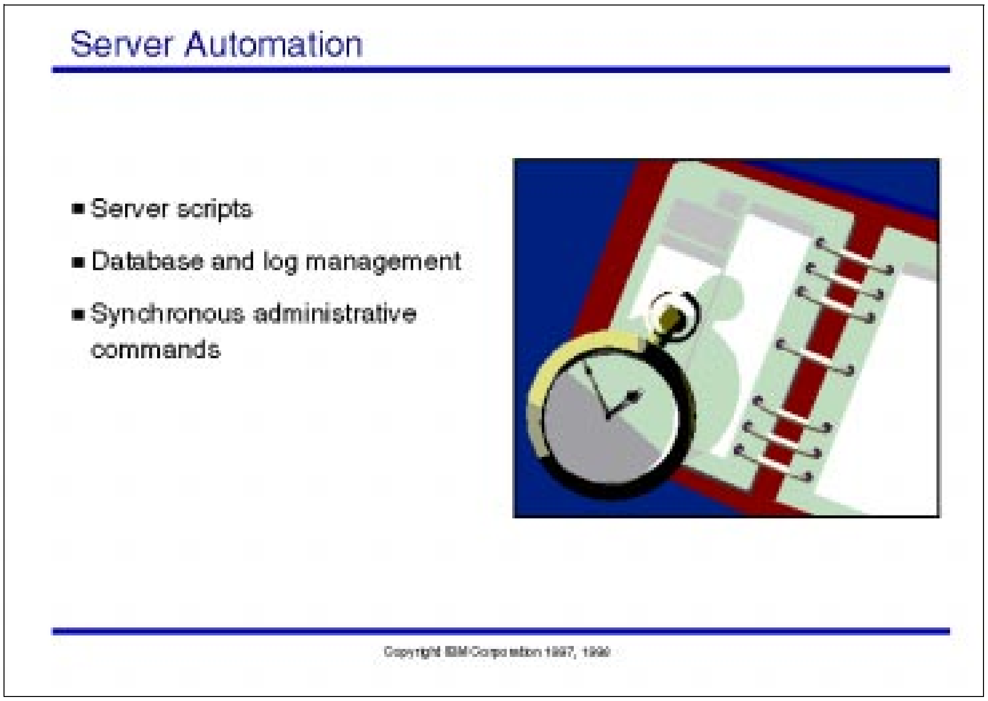
5.2.3.1 Customizable Password Restrictions

In Version 3 an administrator can define a systemwide value for the minimum length required for passwords. The SET MINPWLENGTH command is used to set this value and accepts an integer between 0 and 64. A MINPWLENGTH of 0 indicates that the length of a new password is not to be checked; this is the default value. This forces all new passwords to be of at least the required number of characters. The QUERY STATUS command or the equivalent GUI function can be used to display the minimum password length for the server.

5.2.3.2 Force Password Reset

In Version 3 an administrator can also force an administrator or client node to reset its password at next server access, using the FORCEPWRESET=YES|NO option on the UPDATE NODE|ADMIN and REGISTER NODE|ADMIN commands. When any of these commands is issued with the FORCEPWRESET=YES option, the node or administrator’s password expires at the next login. The client must then reset its password at that time.

5.3 Server Automation



Server Automation

- Server scripts
- Database and log management
- Synchronous administrative commands

Copyright IBM Corporation 1997, 1998

ADSM Version 3 introduces additional features to automate server administration. This section covers the following topics:

- Server scripts

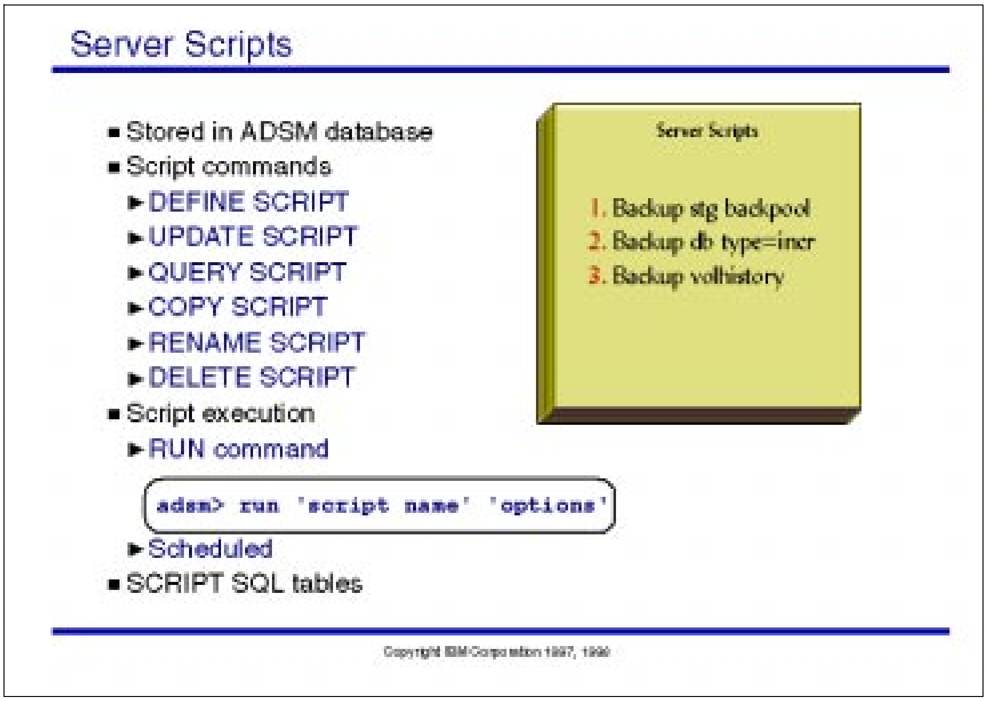
  Server scripts are similar to macros but can now be stored in the database of the ADSM server. The script enhancements include the use of variables, return codes, and script logic. A sample set of scripts is provided with the ADSM server.

- Database and log management

  Additional functions are provided for easy database and log management. These functions allow for automatic extension, and expansion, of the ADSM database or recovery log.

- Synchronous administrative commands

  Administrative commands can now be run in the foreground to facilitate monitoring of server processes. In addition enhanced return codes have been implemented to assist in server automation.



Server Scripts

- Stored in ADSM database
- Script commands
  - DEFINE SCRIPT
  - UPDATE SCRIPT
  - QUERY SCRIPT
  - COPY SCRIPT
  - RENAME SCRIPT
  - DELETE SCRIPT
- Script execution
  - RUN command

```
adsm> run 'script name' 'options'
```

  - Scheduled
- SCRIPT SQL tables

Server Scripts

1. Backup stg backpool
2. Backup db type=incr
3. Backup volhistory

5.3.1 Server Scripts

Scripts are similar to administrative macros. The major difference is that scripts are stored on the ADSM server rather than on the administrative client workstation.

5.3.1.1 Stored in ADSM Database

To enable processing on the ADSM server, scripts are stored in the ADSM database. Each script is referenced by a name of up to 30 characters and may have an additional description of up to 255 characters.

5.3.1.2 Script Commands

New administrative commands are available for creating and managing server scripts:

- The DEFINE SCRIPT command is used to create server scripts. The command is used to initially create the script and the content of the first line within the script:

```
adsm> define script test_script 'query session' line=5
```

This example creates a script named test_script with the QUERY SESSIONS command as line 5 of the script. If a line number is not specified, the command is entered as line 1.

A *file=* option can also be used to specify the name of a text file that contains the administrative commands to be inserted into the script.

- The UPDATE SCRIPT command is used to change or add lines within an existing script. The UPDATE SCRIPT command must specify the line number within the script that is to be added or replaced:

```
adsm> update script test_script 'query process' line=10
```

  This example inserts a new line, line 10, into the previously defined script. Additional lines can be inserted at any free line numbers. If an existing line is specified, it is replaced in the script being updated.

- The QUERY SCRIPT command is used to query defined scripts:

```
adsm> query script test_script format=detail

                          Name: TEST_SCRIPT
                   Line Number: 5
                       Command: query session
Last Update by (administrator): ADMIN
         Last Update Date/Time: 08/06/1998 16:57:09

                          Name: TEST_SCRIPT
                   Line Number: 10
                       Command: query process
Last Update by (administrator): ADMIN
         Last Update Date/Time: 08/06/1998 16:58:45
```

  If the format=detailed option is omitted, only the script name is displayed.

- The COPY SCRIPT command is used to copy the contents of a script to another script with a new name.
- The RENAME SCRIPT command is used to rename an existing script.
- The DELETE SCRIPT command is used to delete a script.

5.3.1.3 Script Execution

As scripts are stored in the ADSM database, they can be accessed and run by any administrator with access to the server.

Scripts are executed by issuing the new RUN command with the script name as a parameter and with any additional options that are required by the commands within the script:

```
adsm> run test_script
```

The RUN command has a *preview* option, which can be used to preview the commands within the script without them being executed. The default for this option is *no,* which will execute the script.

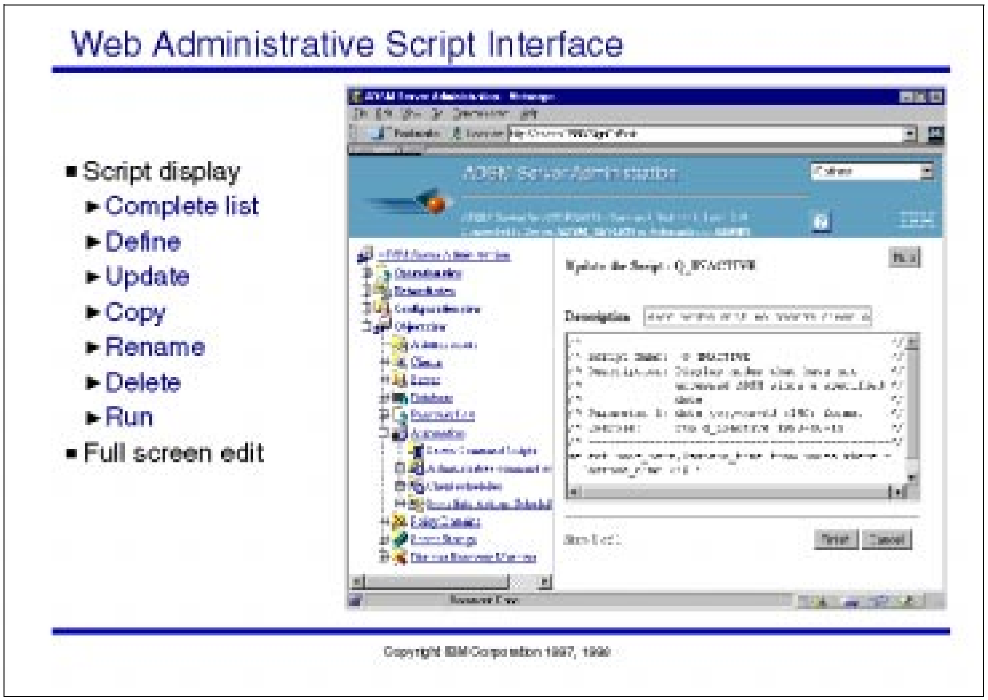
The RUN command, with a script name and options, can also be defined within another server script to create *nested* scripts. However, this cannot be done if the RUN commands create an endless loop condition.

Server scripts can also be executed by administrative schedules on the server.

5.3.1.4 SCRIPT SQL Tables

Administrative SELECT commands can be used to query server scripts. There are two tables: SCRIPT_NAMES and SCRIPTS. The SCRIPT_NAMES table contains the script name, description, and information about when the script was last updated. The SCRIPTS table contains all of the lines defined in the scripts:

```
adsm> select line,command from scripts where name='TEST_SCRIPT'

       LINE     COMMAND
-----------     ------------------
          5     query session
         10     query process
```



Web Administrative Script Interface

- Script display
  - Complete list
  - Define
  - Update
  - Copy
  - Rename
  - Delete
  - Run
- Full screen edit

Copyright IBM Corporation 1997, 1998

5.3.2 Web Administrative Script Interface

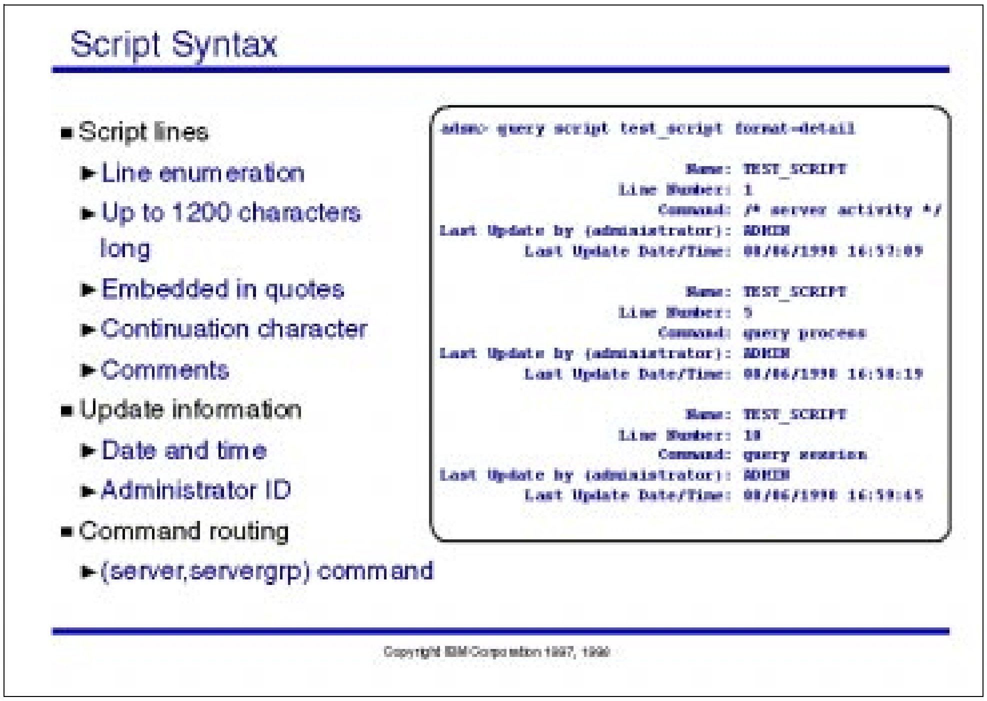
An alternative to commands for managing server scripts is the Web administrative interface.

5.3.2.1 Script Display

The Web administrative interface provides an easy interface for working with server scripts. A script can be selected from a list of defined scripts by clicking on the *Automate operations* hyperlink in either the Operation or Object view. Individual scripts can be selected, and *update*, *copy*, *rename*, *delete,* or *run* options are available.

5.3.2.2 Full Screen Edit

When *update* is selected, a full screen editor is displayed where existing lines can be edited, new lines added, or old lines deleted.



Script Syntax

- Script lines
  - Line enumeration
  - Up to 1200 characters long
  - Embedded in quotes
  - Continuation character
  - Comments
- Update information
  - Date and time
  - Administrator ID
- Command routing
  - (server,servergrp) command

```
adsm> query script test_script format=detail

                      Name: TEST_SCRIPT
               Line Number: 1
                   Command: /* server activity */
Last Update by (administrator): ADMIN
     Last Update Date/Time: 08/06/1998 16:57:09

                      Name: TEST_SCRIPT
               Line Number: 5
                   Command: query process
Last Update by (administrator): ADMIN
     Last Update Date/Time: 08/06/1998 16:58:19

                      Name: TEST_SCRIPT
               Line Number: 10
                   Command: query session
Last Update by (administrator): ADMIN
     Last Update Date/Time: 08/06/1998 16:59:45
```

5.3.3 Script Syntax

A server script consists of one or more lines. Lines within a script have syntax and formatting rules.

5.3.3.1 Script Lines

Each line within a script is a separate entity and is identified by a line number. Line numbers do not have to be consecutive. The first line that can contain commands is line 1. Each line of a script is restricted to a length of 1200 characters. If a line contains blanks, it has to be embedded in quotation marks, for example 'query db'. A script line may contain a command or part of it, a comment, or a description of the script.

Commands can spread over several lines. To continue a command on the next line a continuation character (-) must be used at the end of the line. Comments within command lines are delimited by /* and */.

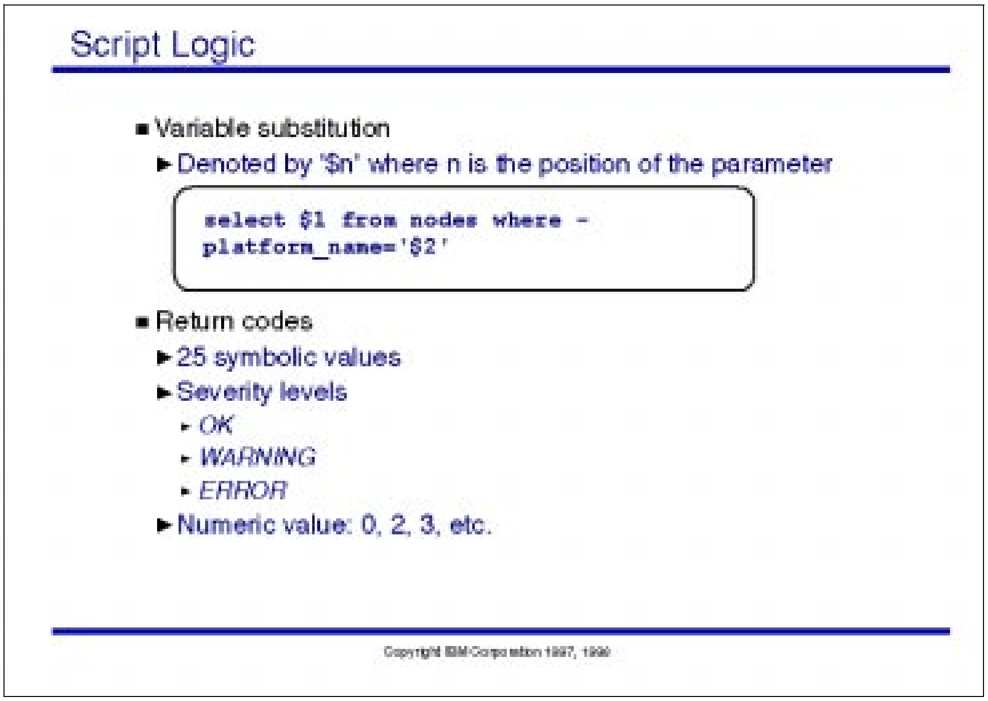
5.3.3.2 Update Information

In addition to the command name, command line contents, and line number, each line of a script is marked with the last date and time of update and with the updating administrative ID.

5.3.3.3 Command Routing

To use command routing (see 6.2, “Administrative Command Routing” on page 190) within server scripts, the following syntax must be used:

```
(routing information) command
```



Script Logic

- Variable substitution
  - Denoted by '$n' where n is the position of the parameter

```
select $1 from nodes where -
platform_name='$2'
```

- Return codes
  - 25 symbolic values
  - Severity levels
    - *OK*
    - *WARNING*
    - *ERROR*
  - Numeric value: 0, 2, 3, etc.

5.3.4 Script Logic

To enhance logic flow, server scripts support variable substitution and enhanced return codes.

5.3.4.1 Variable Substitution

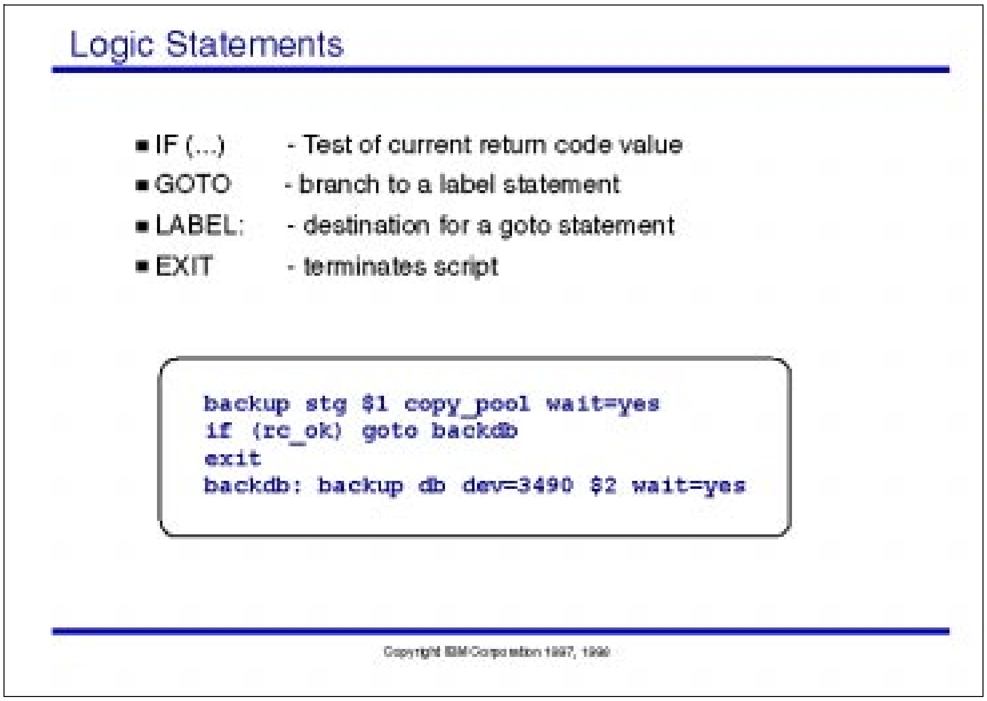
Server scripts can use variable substitution when executed. Values for substituted variables are specified with the RUN command. Placeholders for substitution variables are denoted by $ followed by a number that represents the position location of the variable.

5.3.4.2 Return Codes

Each command that is executed within a server script sets a return code. Each return code has three values: a symbolic value, a severity level, and a numeric value. The symbolic values, which are used within scripts, take the form:

- RC_OK
- RC_NOTFOUND
- RC_UNKNOWN
- RC_SYNTAX
- etc...

Each symbolic value has a corresponding severity level of OK, WARNING, or ERROR, and a numeric value, which is the same as that returned by the administrative command line client.



Logic Statements

- IF (...) - Test of current return code value
- GOTO - branch to a label statement
- LABEL: - destination for a goto statement
- EXIT - terminates script

```
backup stg $1 copy_pool wait=yes
if (rc_ok) goto backdb
exit
backdb: backup db dev=3490 $2 wait=yes
```

5.3.5 Logic Statements

Server scripts also introduce support for conditional logic flow based on return codes. Three commands and a label statement are provided.

5.3.5.1 IF(...)

The IF(...) clause at the beginning of a command line is used to determine how to proceed with processing on the basis of the current return code value (set by the last executed command in the script). Return codes to check are provided as a list in the parentheses of the IF(...) clause. This list may contain return code values and/or severities. If the current return code is equal to any of the values provided in the list, the remainder of the line is executed. If the current return code is not equal to one of the list values, the line is skipped.

5.3.5.2 GOTO

The GOTO statement points to a label (see below). When this statement is encountered, processing will continue with the line starting with the specified label. The line with the specified label has to be a line following the GOTO statement.

5.3.5.3 LABEL:

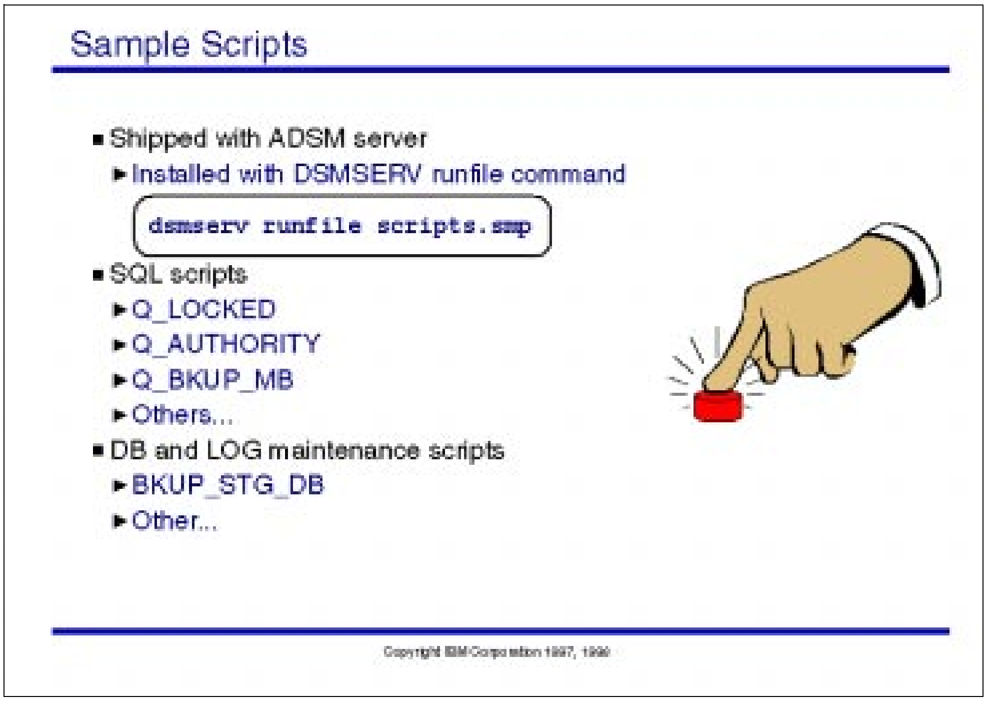
The label statement is provided as the target for the GOTO statement. After the colon, the line can contain a command, or the EXIT statement, or it may be empty. The label is restricted to 30 characters.

5.3.5.4 EXIT

When this statement is encountered, processing for the script terminates.

The following example uses two variables: The first passes a storage pool name to be backed up, the second, parameters to the database backup, for example, TYPE=FULL. The example also shows the use of the IF, EXIT, GOTO, and LABEL: statements.

```
backup stgpool and db full or incremental
backup stg $1 copy_pool wait=yes
if (rc_ok) goto backdb
exit
backdb: backup db dev=3490 $2 wait=yes
```



Sample Scripts

- Shipped with ADSM server
  - Installed with DSMSERV runfile command

```
dsmserv runfile scripts.smp
```

- SQL scripts
  - Q_LOCKED
  - Q_AUTHORITY
  - Q_BKUP_MB
  - Others...
- DB and LOG maintenance scripts
  - BKUP_STG_DB
  - Other...

5.3.6 Sample Scripts

Sample scripts are provided with the ADSM server.

5.3.6.1 Shipped with ADSM Server

A set of 34 sample scripts is shipped with the ADSM Version 3 server. These scripts are not installed by the server installation process. The scripts can be installed by using the following DSMSERV RUNFILE command:

```
dsmserv runfile scripts.smp
```

5.3.6.2 SQL Scripts

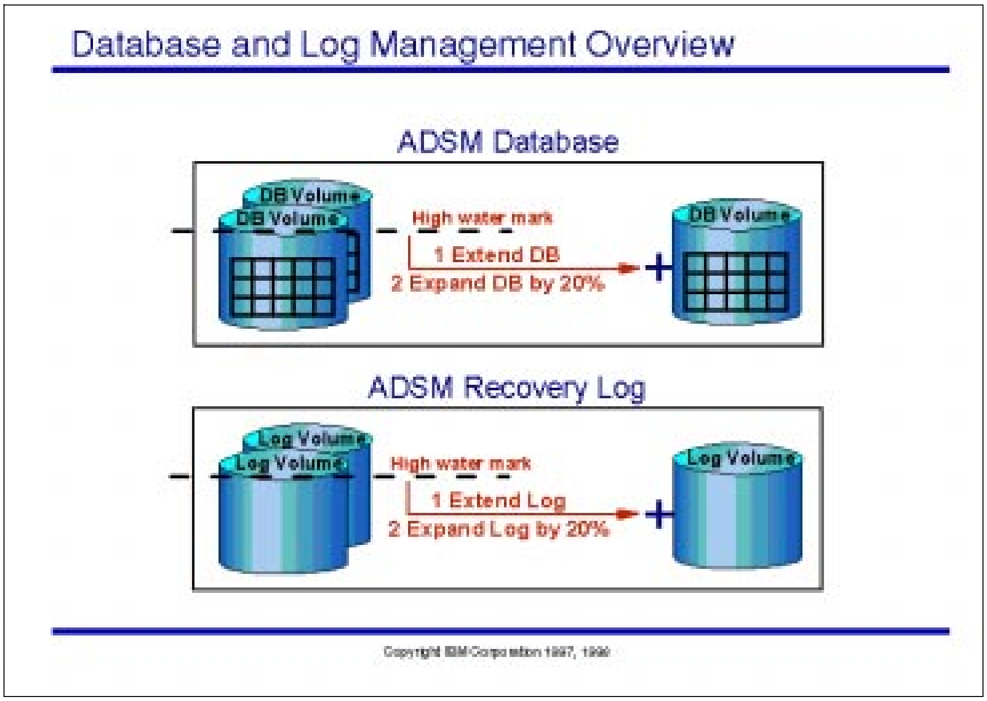
Of the sample scripts 31 are SQL queries. The following are examples of some of the SQL scripts provided:

- Q_LOCKED
  Display locked nodes and administrators
- Q_AUTHORITY
  Display administrators who have a certain authority. A parameter has to be provided to ask for a specific authority in the form x_priv, for example, *Q_AUTHORITY storage_priv.*
- Q_BKUP_MB
  Display nodes that occupy space for backup data on the server exceeding *n* MB. A parameter has to be provided to ask for a certain amount of storage in MB, for example, *Q_BKUP_MB 15.*

5.3.6.3 DB and LOG Maintenance Scripts

Three sample scripts are provided for database and log maintenance tasks:

- BKUP_STG_DB
  This script backs up storage pools to copy storage pools, backs up the database if the storage pool backup was successful, and reschedules itself again 2 hr later.
- DEF_DB_EXTEND
  This script defines a previously allocated database volume and, if successful, extends the database.
- DEF_LOG_EXTEND
  This script defines a previously allocated log volume and, if successful, extends the log.



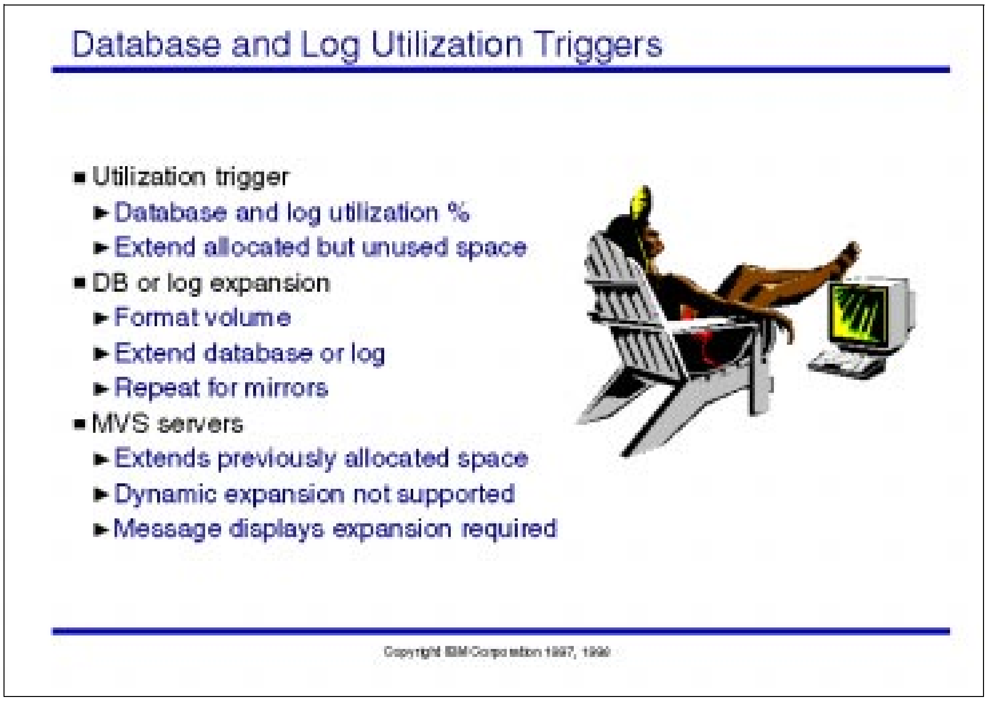
Database and Log Management Overview

5.3.7 Database and Log Management Overview

A new feature for ADSM server database and recovery log management helps ensure continuous operation of an ADSM server. The objective of this feature is to prevent database or log out-of-space situations from affecting server availability.

An ADSM administrator can define a utilization threshold for the database or recovery log. When this threshold is reached, the server first attempts to extend to the database or log, using space previously defined but not used in database or log volumes.

If sufficient space is not available to extend the database and reduce the utilization below the threshold, the database or log is expanded. To expand the database or log, the ADSM server allocates additional database or log volumes, then extends the database or log to use this newly allocated space. Expansion is performed on the basis of a percentage of the current database or log size.



Database and Log Utilization Triggers

- Utilization trigger
  - Database and log utilization %
  - Extend allocated but unused space
- DB or log expansion
  - Format volume
  - Extend database or log
  - Repeat for mirrors
- MVS servers
  - Extends previously allocated space
  - Dynamic expansion not supported
  - Message displays expansion required

5.3.8 Database and Log Utilization Triggers

Automatic database and log extension and expansion are based on utilization triggers.

5.3.8.1 Utilization Trigger

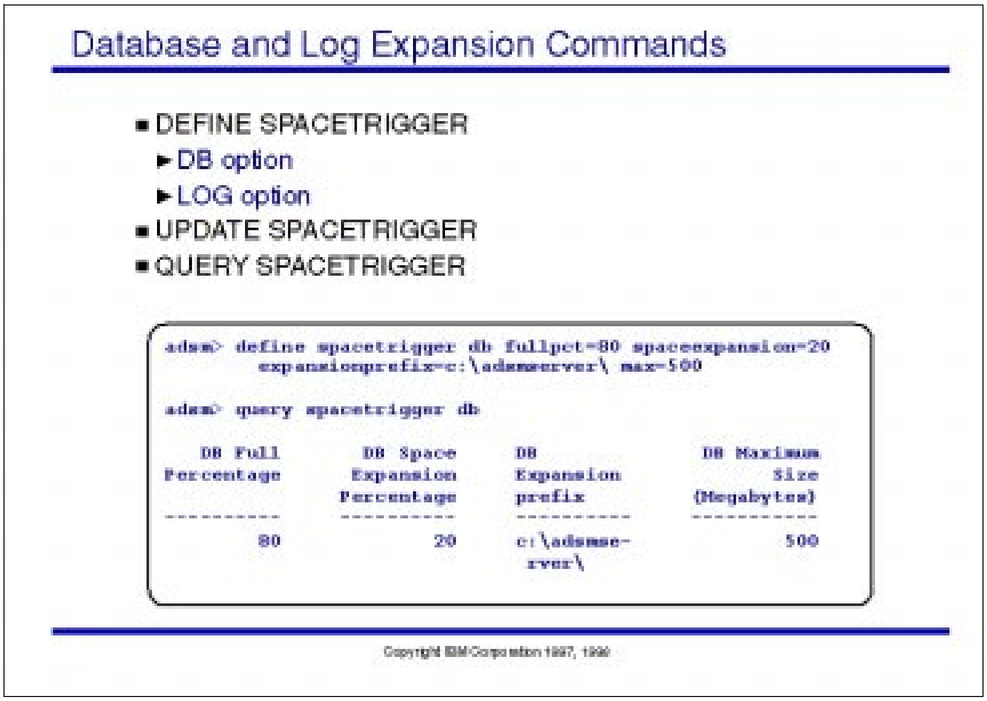
Administrator-defined space triggers tell the ADSM server when to increase the size of the database or log. Space triggers define a maximum utilization percentage. When that utilization percentage is reached, the first step is to increase the size of the database or log, using space previously allocated but not used. This is done by *extending* the database.

5.3.8.2 DB or Log Expansion

If there is no previously allocated space, or insufficient space to reduce the utilization percentage below the trigger value, the database or log is *expanded*. The first step in expanding the database is to allocate a new volume. A volume prefix name and the amount of space by which to extend as a percentage of the existing database or log have to be predefined. A constraining upper limit can be defined for the database or log to prevent extension beyond a certain point. After allocation the volumes are formatted. When formatting is finished, the database or log will be extended by the size of the newly allocated volumes. Mirrored volumes will be allocated as well, if the database or log is mirrored.

5.3.8.3 MVS Servers

The MVS server partially supports these new functions. Automatic extension using existing space is supported. Database expansion using MVS dynamic data set allocation is not supported. Instead, a message displaying the required expansion will be issued.



Database and Log Expansion Commands

- DEFINE SPACETRIGGER
  - DB option
  - LOG option
- UPDATE SPACETRIGGER
- QUERY SPACETRIGGER

```
adsm> define spacetrigger db fullpct=80 spaceexpansion=20
        expansionprefix=c:\adsmserver\ max=500

adsm> query spacetrigger db

   DB Full        DB Space     DB               DB Maximum
Percentage       Expansion     Expansion              Size
                Percentage     prefix          (Megabytes)
----------      ----------     ----------      -----------
        80              20     c:\adsmse-              500
                                rver\
```

Copyright IBM Corporation 1997, 1999

5.3.9 Database and Log Expansion Commands

New commands are introduced for automatic database and log expansion.

5.3.9.1 DEFINE SPACETRIGGER

The DEFINE SPACETRIGGER command defines settings for the database and recovery log space triggers. These triggers determine when and how ADSM deals with space shortages in the database and recovery log. The following parameters can be used:

DB/LOG Positional parameter, specifies whether the database or log is to be managed by automatic extension

Fullpct=percentage Specifies when ADSM automatically allocates more space for the database or recovery log. When this value is reached, ADSM attempts to acquire more space if space expansion is specified. This parameter is optional.

SPACEexpansion=percentage Specifies the expansion size of the database or recovery log in percent of the existing size. ADSM creates new volumes in multiples of 9 MB to achieve the required size.

EXPansionprefix=prefixname Specifies the prefix that ADSM will use in creating files to expand the database and the recovery log. The prefix name can include one or more directory separator characters, such as EXPansionprefix=j:\adsmserv\. ADSM does not check the prefix name for validity. If an incorrect prefix is specified, automatic expansion can fail.

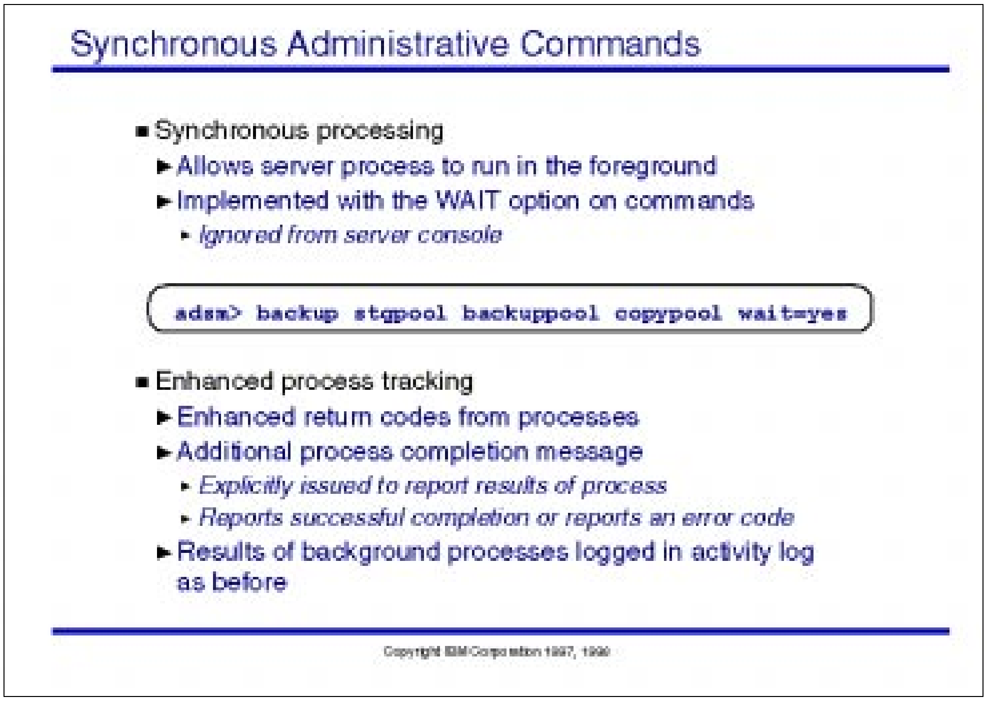
MAXimumsize=number Specifies the maximum size of the database or log. If the current database or log is below this size when the utilization trigger is reached, the expansion will proceed, even if the expanded size exceeds this maximum size setting. Subsequent expansions will then fail because this size was exceeded. This parameter is optional.

5.3.9.2 UPDATE SPACETRIGGER

This command allows previously defined parameters to be changed for automatic expansion.

5.3.9.3 QUERY SPACETRIGGER

This command displays the current settings for automatic database expansion.



Synchronous Administrative Commands

- Synchronous processing
  - Allows server process to run in the foreground
  - Implemented with the WAIT option on commands
    - *Ignored from server console*

```
adsm> backup stgpool backuppool copypool wait=yes
```

- Enhanced process tracking
  - Enhanced return codes from processes
  - Additional process completion message
    - *Explicitly issued to report results of process*
    - *Reports successful completion or reports an error code*
  - Results of background processes logged in activity log as before

5.3.10 Synchronous Administrative Commands

In Version 2 many commands start as background processes because of the amount of time they take to complete. Once a process has been started in the background within ADSM, the status of the process can be obtained with the QUERY PROCESS command. However, once the process has finished, the only way to determine the outcome of that process is to query the activity log to look for process start and completion and any related messages.

5.3.10.1 Synchronous Processing

In Version 3 a new parameter has been added to many commands that in Version 2 would have started background processes, to force them to run in the foreground and concurrently report any messages including a message and return code on completion. The WAIT=YES option runs the process in the foreground, reports messages concurrently, and returns to the issuing session only on completion. Thus an administrator can issue a command and have real-time reporting on its progress along with a meaningful message on completion.

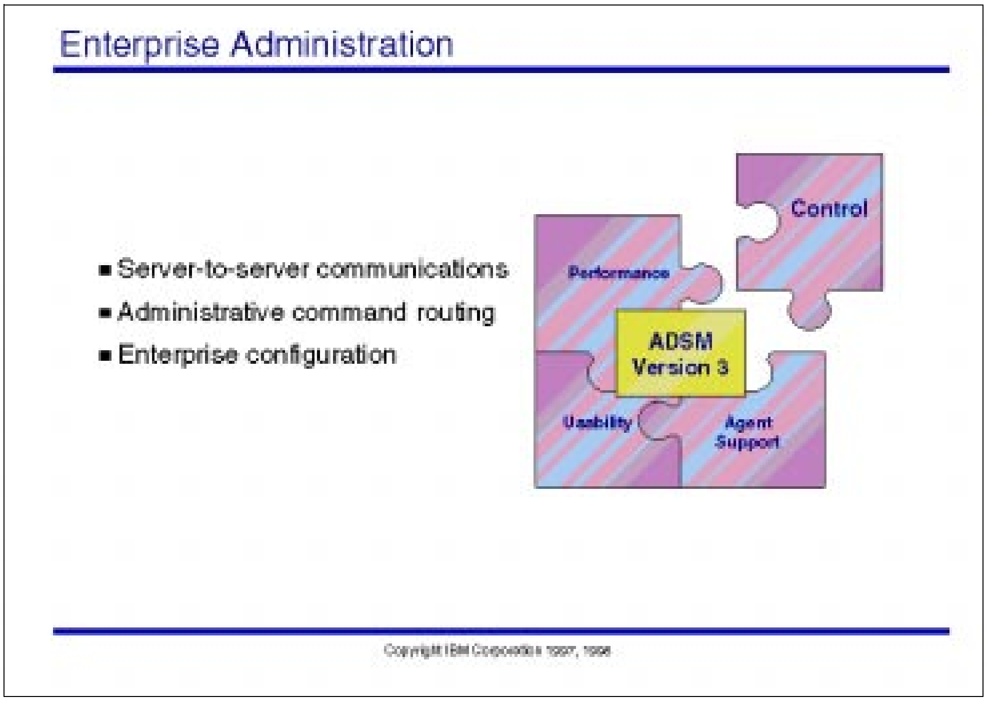
5.3.10.2 Enhanced Process Tracking

To improve process tracking in general, changes have been made in Version 3 to provide enhanced process return codes from both foreground and background operations. The message reported at process completion has also been improved to explicitly state the message, the command, and its success or failure in the form of either a SUCCESS or ERROR message:

```
adsm> backup stgpool backuppool copypool wait=yes
ANR0984I Process 24 for BACKUP STORAGE POOL started
         in the FOREGROUND at 16,55,20.
ANR1210I Backup of primary storage pool BACKUPPOOL to
         copy storage pool COPYPOOL started as process 24.
ANR1212I Backup process 24 ended for storage pool BACKUPPOOL.
ANR0986I Process 24 for BACKUP STORAGE POOL running in the
         FOREGROUND processed 23 items for a total of
         2,863,104 bytes with a completion state of SUCCESS at 16,56,05
ANR1214I Backup of primary storage pool BACKUPPOOL to
         copy storage pool COPYPOOL has ended.  Files Backed Up: 23,
         Bytes Backed Up: 2863104,
         Unreadable Files: 0, Unreadable Bytes: 0.
adsm>
```

The reporting mechanism is the same when processes are allowed to run in the background, except that the messages are logged in the activity log.

Chapter 6. Enterprise Administration



Enterprise Administration

- Server-to-server communications
- Administrative command routing
- Enterprise configuration

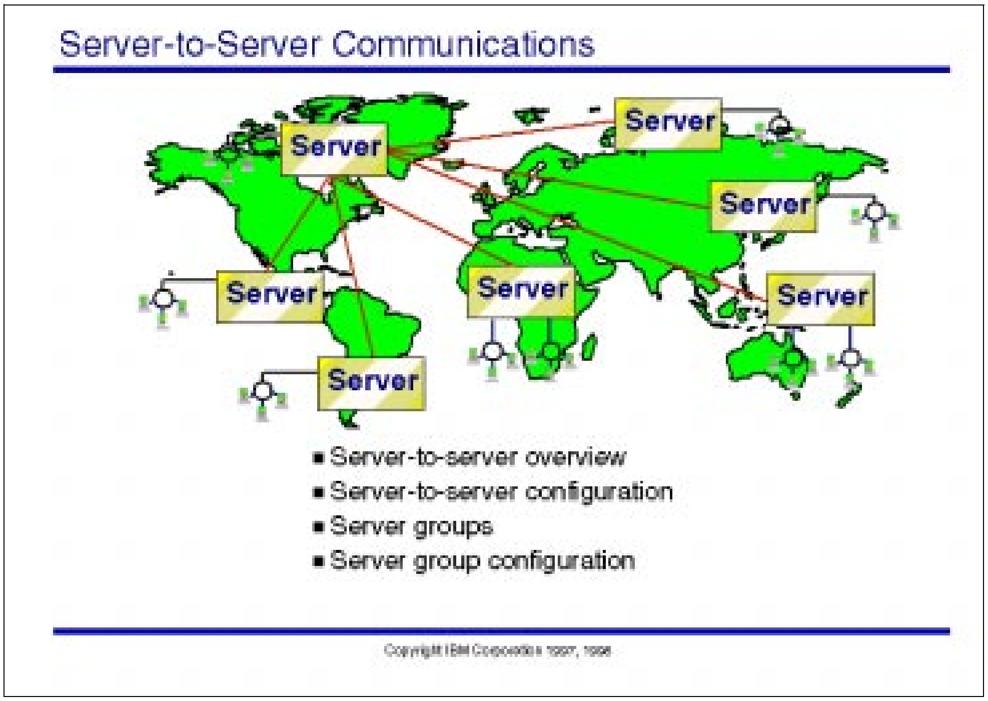
Control
Performance
ADSM Version 3
Usability
Agent Support

Copyright IBM Corporation 1997, 1998

This chapter describes the ADSM Version 3 features for administering multiple ADSM servers in an enterprise.The following topics are covered:

- Server-to-server communications
- Administrative command routing
- Enterprise configuration

6.1 Server-to-Server Communications



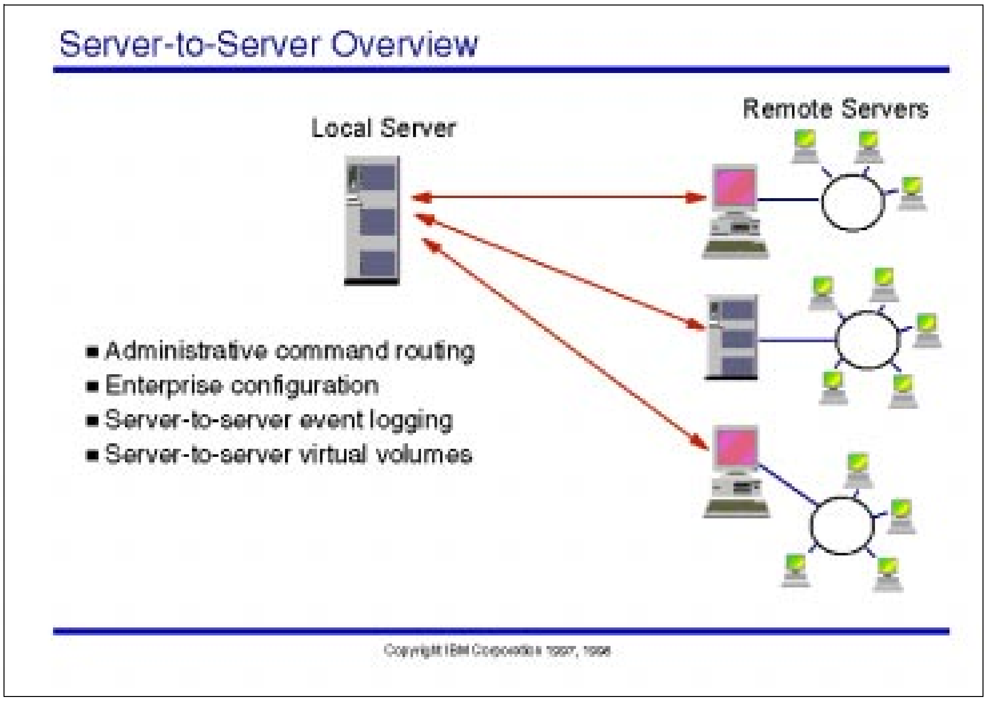
Server-to-Server Communications

- Server-to-server overview
- Server-to-server configuration
- Server groups
- Server group configuration

Copyright IBM Corporation 1997, 1998

ADSM Version 3 enables multiple ADSM servers within an enterprise to be configured and administered from a central location. ADSM Version 3 server-to-server communications provides the foundation for configuring multiple ADSM Version 3 servers in an enterprise. This section covers the following topics:

- Server-to-server overview
- Server-to-server configuration
    - Remote server setup
    - Local server setup
    - Server definitions
- Server groups
- Server group configuration



Server-to-Server Overview

- Administrative command routing
- Enterprise configuration
- Server-to-server event logging
- Server-to-server virtual volumes

Copyright IBM Corporation 1997, 1998

6.1.1 Server-to-Server Overview

Server-to-server communications provides a mechanism for creating network connections between ADSM Version 3 servers. Server-to-server communications is enabled by a series of definitions made on the servers. In the configuration examples we use the following terminology:

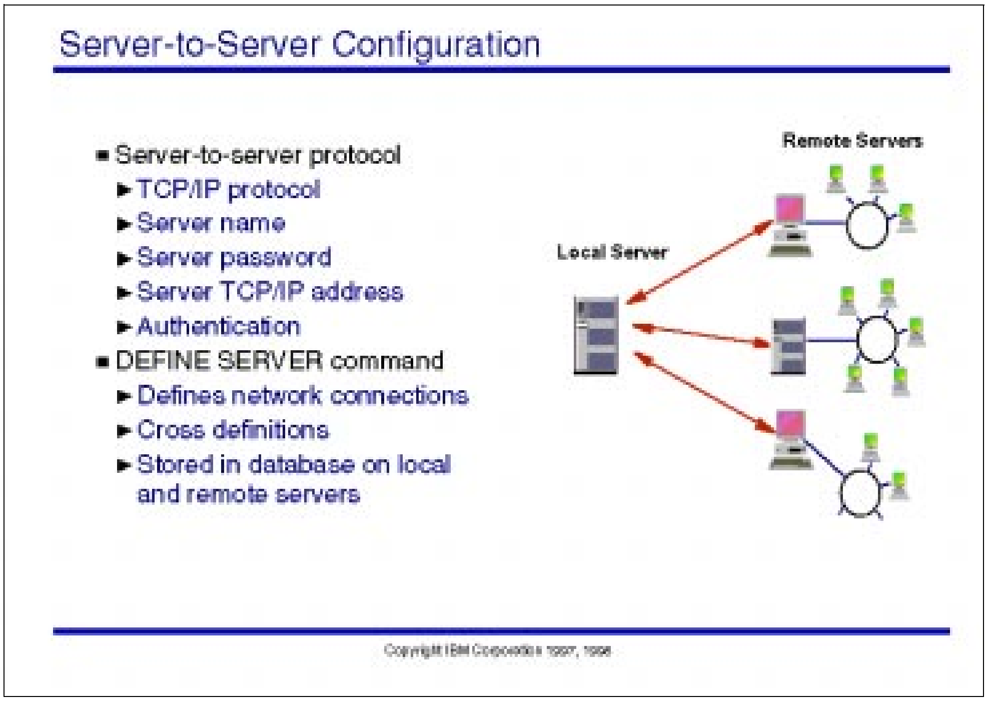
- *Local server* - The server on which server definitions are created
- *Remote server*s - Servers defined by the local server

Configuring server-to-server communications is a prerequisite for the following ADSM Version 3 functions:

- Administrative command routing
- Enterprise configuration
- Server-to-server virtual volumes
- Server-to-server event logging

Server Topology

The server definitions and terms used in this example are used to illustrate a simple hierarchical model of a single local, or central, server and three remote, or distributed, servers. Server-to-server configurations are not limited to such a hierarchical model. Any combination of server definitions can be established.



Server-to-Server Configuration

- Server-to-server protocol
  - TCP/IP protocol
  - Server name
  - Server password
  - Server TCP/IP address
  - Authentication
- DEFINE SERVER command
  - Defines network connections
  - Cross definitions
  - Stored in database on local and remote servers

6.1.2 Server-to-Server Configuration

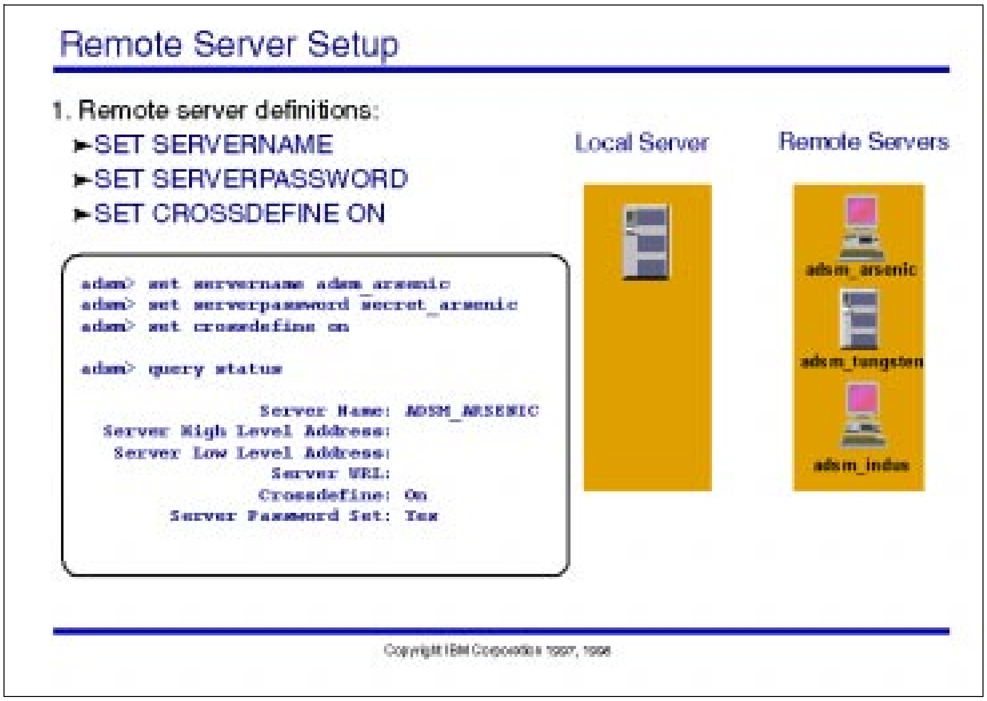
Server-to-server communications is based on a TCP/IP connection between the local and managed servers. Server-to-server communications is configured with a series of definitions on the local and remote servers.

6.1.2.1 Server-to-Server Protocol

Server-to-server communications uses TCP/IP as the communication protocol between servers. Servers are defined with a server name, a server password, and the server TCP/IP address. The server name and password are used to authenticate sessions being initiated by the local server. There are no password prompts when sessions are initiated between servers.

6.1.2.2 DEFINE SERVER Command

The DEFINE SERVER command is used to define the network connection between the local and remote servers. The command, issued on the local server, specifies the server name, server password, and TCP/IP address of the remote server being defined. A *crossdefine* function can be enabled on the remote server being defined. The remote server receives the server name, server password, and TCP/IP address of the local server performing the server definition. When the crossdefine function is enabled on the remote server, the remote server receives this local server configuration information and dynamically creates a reciprocal server definition for the local server on the remote server. This function reduces the number of server definitions required by the administrator.



Remote Server Setup

1. Remote server definitions:
   - ►SET SERVERNAME
   - ►SET SERVERPASSWORD
   - ►SET CROSSDEFINE ON

```
adsm> set servername adsm_arsenic
adsm> set serverpassword secret_arsenic
adsm> set crossdefine on

adsm> query status

                Server Name: ADSM_ARSENIC
  Server High Level Address:
   Server Low Level Address:
                 Server URL:
                Crossdefine: On
        Server Password Set: Yes
```

Local Server
Remote Servers
adsm_arsenic
adsm_tungsten
adsm_indus

Copyright IBM Corporation 1997, 1998

6.1.3 Remote Server Setup

This configuration example consists of a local server and three remote servers. The example only illustrates the definitions for one of the remote servers as the configuration steps are repeated for the other remote servers. Configuration consists of three steps:

1. Remote server setup
2. Local server setup
3. Server definitions

The first step is to set up the remote server so that it can be defined on the local server and to enable it to perform a server crossdefine operation back to the local server.

6.1.3.1 SET SERVERNAME

Server-to-server communications use the ADSM server name to identify the server. The SETSERVERNAME command is used to set an appropriate server name. All servers should be identified with unique server names. By default the server name is set to ADSM following server installation. The server name set on the remote server will be used when the server-to-server definitions are made on the local server. The server name is also displayed whenever a backup-archive client starts a session and on the Web administrative central console.

6.1.3.2 SET SERVERPASSWORD

The server password is used to authenticate sessions initiated from other servers. The SET SERVERPASSWORD command is used to define the server password. Once defined, the server password cannot be displayed by an administrator. The server password can be reset by reissuing the SET SERVERPASSWORD command with a new password. The server password set on the remote server will also be used when the server-to-server definitions are made on the local server. The server name and password are used to authenticate sessions from other servers.

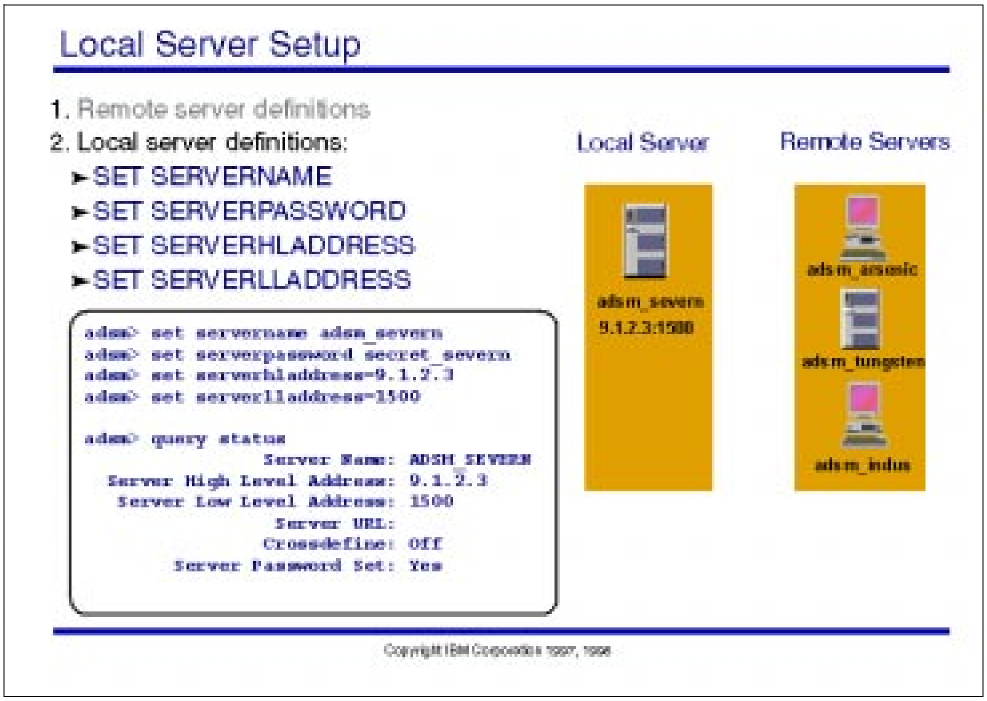
6.1.3.3 SET CROSSDEFINE ON

The SET CROSSDEFINE ON command enables the remote server to automatically create a server definition for the local server. Crossdefine reduces the number of server definitions administrators must create. When crossdefine is used, each remote server, when defined on the local server, automatically configures a definition for the local server on the remote server. The alternative to using crossdefine is to define each remote server to the local server and then separately define the local server on each of the remote servers.

The server name, crossdefine option, and server password status can be viewed by using the QUERY STATUS command on the remote server:

```
adsm> query status
ADSM Server for Windows NT - Version 3, Release 1, Level 2.0

                                Server Name: ADSM_ARSENIC
                  Server High Level Address:
                   Server Low Level Address:
                                 Server URL:
                                Crossdefine: On
                        Server Password Set: Yes
```

The Yes value in the *Server Password Set:* field indicates that the server password has been set.



Local Server Setup

1. Remote server definitions
2. Local server definitions:
   - ►SET SERVERNAME
   - ►SET SERVERPASSWORD
   - ►SET SERVERHLADDRESS
   - ►SET SERVERLLADDRESS

```
adsm> set servername adsm_severn
adsm> set serverpassword secret_severn
adsm> set serverhladdress=9.1.2.3
adsm> set serverlladdress=1500

adsm> query status
                    Server Name: ADSM_SEVERN
  Server High Level Address: 9.1.2.3
   Server Low Level Address: 1500
                     Server URL:
                    Crossdefine: Off
           Server Password Set: Yes
```

6.1.4 Local Server Setup

The second step is to set up the local server so that it can be defined on the remote server when the server crossdefine operation is performed by the remote server.

The following four server definitions on the local server are passed to the remote server and used by that server to perform the crossdefine operation of the local server on the remote server.

6.1.4.1 SET SERVERNAME

The SETSERVERNAME command is used to set the server name on the local server. This server name will be passed to the remote server when the local server creates the server-to-server definitions.

6.1.4.2 SET SERVERPASSWORD

The SET SERVERPASSWORD command is used to define the local server password. This server password will also be passed to the remote server when the local server creates the server-to-server definitions.

6.1.4.3 SET SERVERHLADDRESS

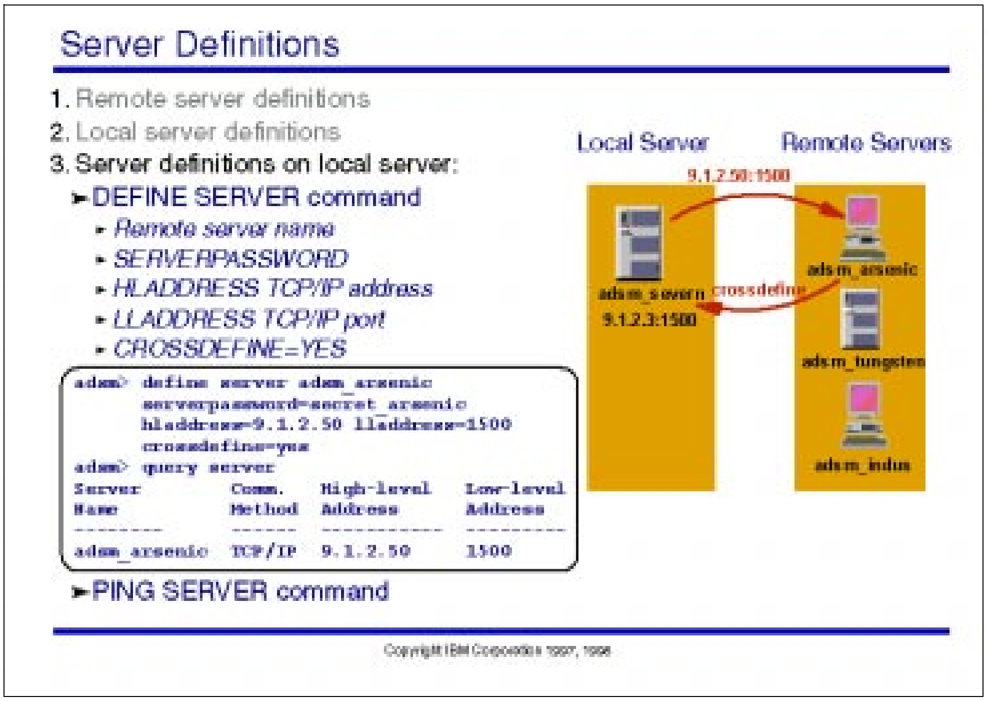
The SET SERVERHLADDRESS command is used to specify the TCP/IP address of the local server in dotted decimal format. This address will also be passed to the remote server when the local server creates the server-to-server definitions.

6.1.4.4 SET SERVERLLADDRESS

The SET SERVERLLADDRESS command is used to specify the TCP/IP port number of the local server. This is the TCP/IP port where the local server listens for incoming TCP/IP connections from other servers or clients. This port number will also be passed to the remote server when the local server creates the server-to-server definitions.

The QUERY STATUS command can be issued to display the settings:

```
adsm> query status
ADSM Server for AIX-RS/6000 - Version 3, Release 1, Level 2.0

                                Server Name: ADSM_SEVERN
                  Server High Level Address: 9.1.2.3
                   Server Low Level Address: 1500
                                 Server URL:
                                Crossdefine: Off
                        Server Password Set: Yes
```



Server Definitions

1. Remote server definitions
2. Local server definitions
3. Server definitions on local server:
   - DEFINE SERVER command
     - *Remote server name*
     - *SERVERPASSWORD*
     - *HLADDRESS TCP/IP address*
     - *LLADDRESS TCP/IP port*
     - *CROSSDEFINE=YES*

```
adsm> define server adsm_arsenic
      serverpassword=secret_arsenic
      hladdress=9.1.2.50 lladdress=1500
      crossdefine=yes
adsm> query server
Server          Comm.    High-level     Low-level
Name            Method   Address        Address
---------       ------   ------------   ---------
adsm_arsenic    TCP/IP   9.1.2.50       1500
```

   - PING SERVER command

6.1.5 Server Definitions

The third and final step is to define the remote servers on the local server. With the crossdefine option set on the remote server, and the local server setup complete, this step results in a server definition for the remote server on the local server, and a server definition for the local server on the remote server.

6.1.5.1 DEFINE SERVER Command

The DEFINE SERVER command is used on the local server to define a remote server. The following options must be specified:

Remote server name - The name of the remote server. This name should match the server name created with the SET SERVERNAME command on the remote server.

SERVERPASSWORD - The password for the remote server to be defined. The server password specified must match the password set on the remote server with the SET SERVERPASSWORD command.

HLADDRESS - The TCP/IP address of the remote server in dotted decimal format

LLADDRESS - The TCP/IP port number of the remote server

The following parameter is optional:

CROSSDEFINE - Specifies that the local server running this command is also defined on the remote server being defined with this DEFINE SERVER command. The remote server must have set CROSSDEFINE=ON for this to work.

A successful DEFINE SERVER command results in two server definitions. The remote server is defined on the local server, and the local server is defined on the remote.

DEFINE SERVER Passwords

With ADSM Version 3.1.2, the DEFINE SERVER command has changed. Two separate passwords can now be defined. The SERVERPASSWORD is the password used to authenticate server-to-server sessions as described above. There is also a PASSWORD option, which is used to specify a separate password for use with server-to-server virtual volumes (see 8.1, “Server-to-Server Virtual Volumes” on page 242). These two passwords are independent of each other.

The QUERY SERVER command can be issued on either the local server or remote servers to display details of the defined servers:

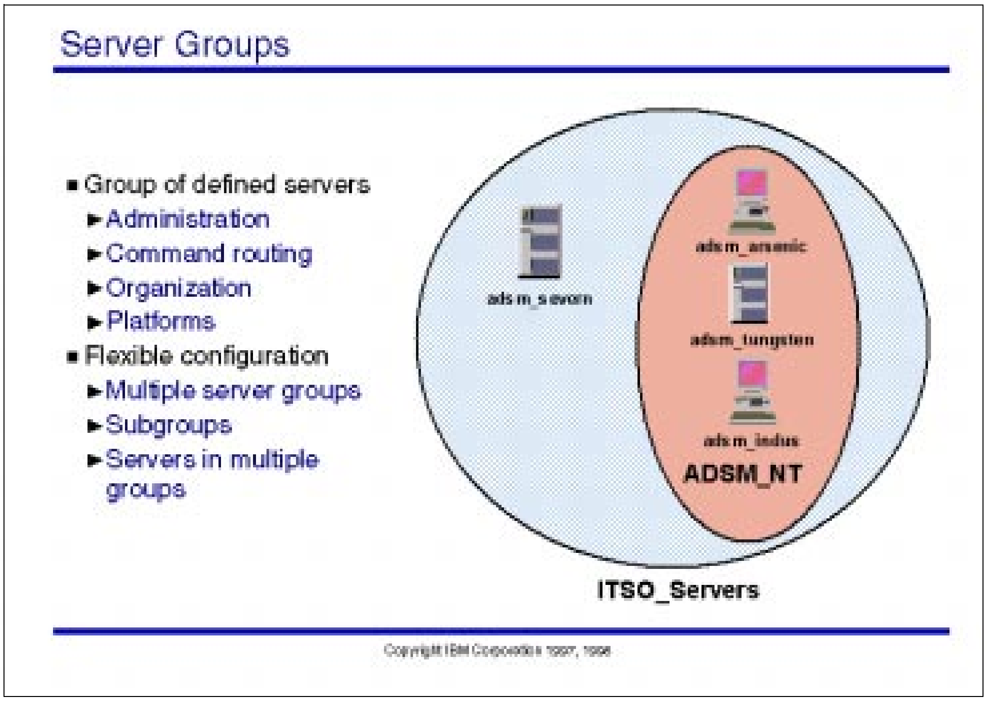
```
adsm> query server adsm_arsenic f=d

                           Server Name: ADSM_ARSENIC
                          Comm. Method: TCPIP
                    High-level Address: 9.1.2.50
                     Low-level Address: 1500
                           Description:
                     Allow Replacement: No
                             Node Name:
                 Last Access Date/Time: 07/01/1998 18:13:35
                Days Since Last Access: 16
                               Locked?: No
                           Compression: No
               Archive Delete Allowed?: (?)
                                   URL:
                Registration Date/Time: 07/01/1998 18:13:35
             Registering Administrator: TIMRES1
           Bytes Received Last Session: 1,747
               Bytes Sent Last Session: 126
         Duration of Last Session (sec): 3.22
           Pct. Idle Wait Last Session: 95.00
          Pct. Comm. Wait Last Session: 0.03
          Pct. Media Wait Last Session: 0.00
                 Grace Deletion Period: 5
                      Managing profile:
                   Server Password Set: Yes
         Server Password Set Date/Time: 07/01/1998 18:13:35
         Days Since Server Password Set: 16
       Invalid Sign-on Count for Server: 0
           Virtual Volume Password Set: No
  Virtual Volume Password Set Date/Time: (?)
 Days Since Virtual Volume Password Set: (?)
Invalid Sign-on Count for Virtual Volume Node: 0
```

6.1.5.2 PING SERVER Command

The PING SERVER command can be used to verify that a defined server can be contacted and that the server passwords are valid:

```
adsm> ping server adsm_arsenic
ANR1706I Ping for server 'ADSM_ARSENIC' was able to establish a connection.
```

The administrator ID used to issue the PING SERVER command must also be registered on the server being pinged. The command starts an administrative session on the server being pinged, using the administrative ID that issues the command. If the ID is not registered, the PING SERVER command will fail.



Server Groups

- Group of defined servers
  - Administration
  - Command routing
  - Organization
  - Platforms
- Flexible configuration
  - Multiple server groups
  - Subgroups
  - Servers in multiple groups

Copyright IBM Corporation 1997, 1998

6.1.6 Server Groups

ADSM servers defined using server-to-server communications can be logically grouped together for administrative purposes. A server group eases server administration where repetitive administration tasks must be performed on multiple ADSM servers. Defining a server group allows an administrator to perform a single administrative task that will be executed on all ADSM servers defined within the server group.

6.1.6.1 Group of Defined Servers

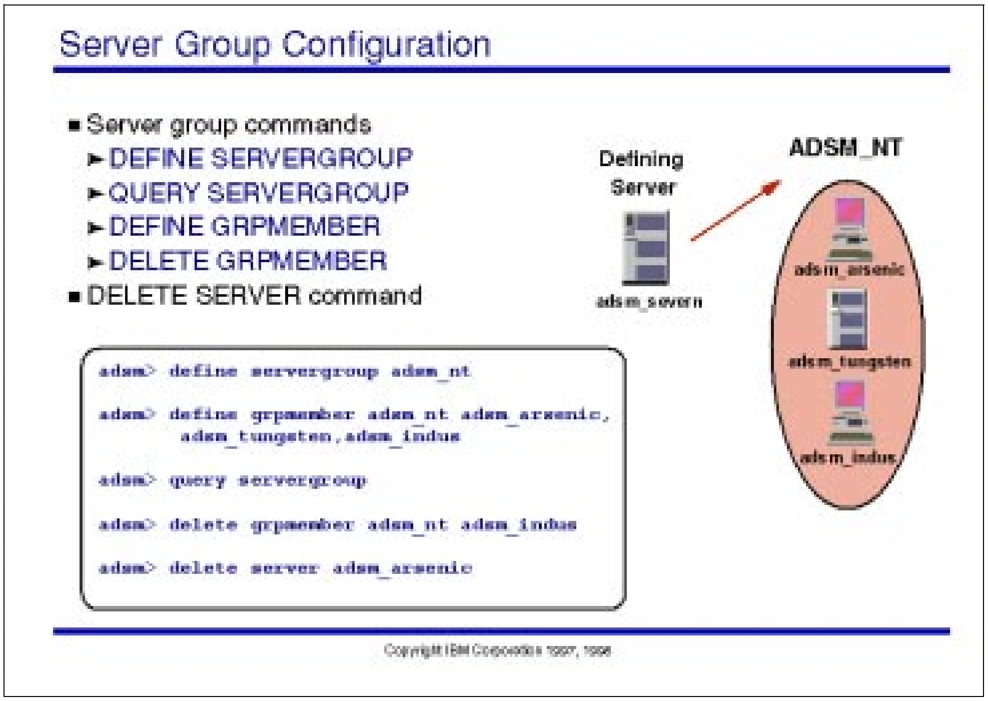
A server group is a collection of defined ADSM servers that have common administrative requirements. The server groups could be organized by business organization, operating system, or any required combination.

With the definition of servers and identical administrators on all of the defined servers, administrative commands can be routed to multiple servers (see “Administrative Command Routing” on page 190).

6.1.6.2 Flexible Configuration

Definition of server groups is very flexible. Any server or server group can be added to any other server group. The following can be defined:

- Multiple server groups
- Groups within groups
- Individual servers defined in more than one server group



Server Group Configuration

- Server group commands
  - DEFINE SERVERGROUP
  - QUERY SERVERGROUP
  - DEFINE GRPMEMBER
  - DELETE GRPMEMBER
- DELETE SERVER command

```
adsm> define servergroup adsm_nt

adsm> define grpmember adsm_nt adsm_arsenic,
       adsm_tungsten,adsm_indus

adsm> query servergroup

adsm> delete grpmember adsm_nt adsm_indus

adsm> delete server adsm_arsenic
```

6.1.7 Server Group Configuration

A server group can be defined on a server regardless of other server definitions. Servers are added to a server group after the server group is defined.

6.1.7.1 Server Group Commands

There are a number of new commands for defining, updating, and managing server groups and the servers defined within the groups:

- DEFINE SERVERGROUP

  This command defines a new server group on the ADSM server. The positional parameter, group_name, defines the name of the server group to be defined.

- DEFINE GRPMEMBER

  This command defines one or more servers as members of a server group. The members to be added can be specified as a list of server names separated by commas with no spaces. The maximum length of any server or group name is 64 characters.

  The server must be defined with the DEFINE SERVER command before it can be added to a server group.

- QUERY SERVERGROUP

  This command displays one or more server groups and the servers defined within the group:

```
dsm> query servergroup adsm_nt

Server Group     Members          Description             Managing profile
-------------    -------------    --------------------    --------------------
ADSM_NT           ADSM_ARSENIC
                  ADSM_INDUS
                  ADSM_TUNGST-
                  EN
```

- DELETE GRPMEMBER

  This command deletes one or more servers as members from a server group. The members to be deleted may be specified as a list of server names separated by commas with no spaces.

There are additional server group commands for managing server group definitions: COPY SERVERGROUP, UPDATE SERVERGROUP, RENAME SERVERGROUP, and DELETE SERVERGROUP. An additional server group member command, MOVE GRPMEMBER, can be used to move a server from one server group to another.

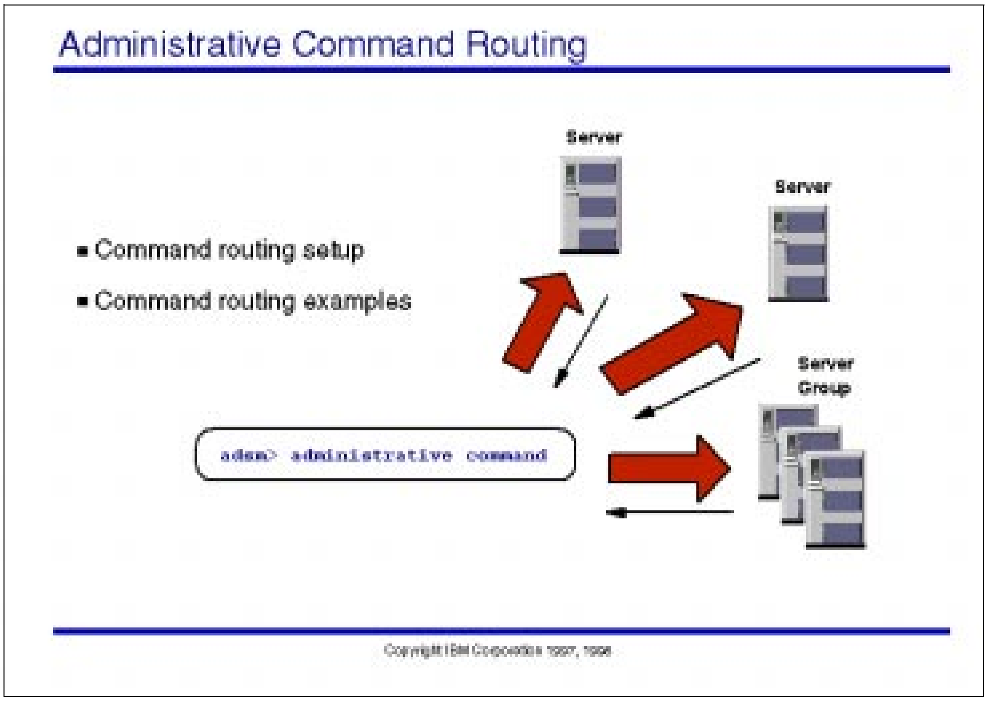
6.1.7.2 DELETE SERVER Command

The DELETE SERVER command deletes the definition of a server configured for server-to-server communications:

```
adsm> delete server ADSM_ARSENIC
```

This command also deletes group member definitions for this server in any server group in which it has been defined.

6.2 Administrative Command Routing



Administrative Command Routing

- Command routing setup
- Command routing examples

Server

Server

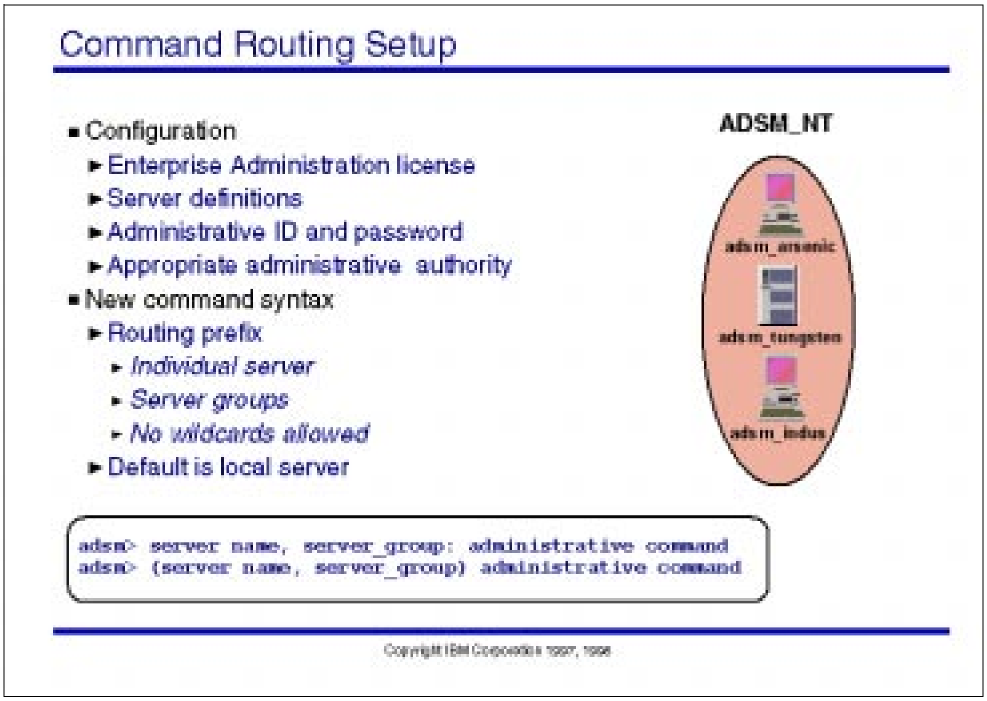
Server Group

```
adsm> administrative command
```

Copyright IBM Corporation 1997, 1998

Administrative command routing enables an administrator to send commands from one ADSM server to one or more other ADSM servers that the administrator is authorized to access. The output of the command is displayed at the server where the command was entered. This section covers the following topics:

- Command routing setup
- Command routing examples



Command Routing Setup

- Configuration
  - Enterprise Administration license
  - Server definitions
  - Administrative ID and password
  - Appropriate administrative authority
- New command syntax
  - Routing prefix
    - *Individual server*
    - *Server groups*
    - *No wildcards allowed*
  - Default is local server

```
adsm> server name, server_group: administrative command
adsm> (server name, server_group) administrative command
```

Copyright IBM Corporation 1997, 1998

6.2.1 Command Routing Setup

To route administrative commands to multiple servers the servers must first be configured. After the servers are configured, a new administrative command syntax is used to route commands to one or more of them. Routed commands are executed on the defined servers, and the command output is returned and formatted on the server where the commands were issued.

6.2.1.1 Configuration

The following configuration must be performed to enable administrative command routing:

- The Enterprise Administration license must be enabled on the server where the administrative commands are issued.
- Crossdefined server definitions must exist between the server where the commands will be issued and the servers where the commands are routed.
- The administrative ID and password on the server issuing the commands must also be valid on the server where the commands are to be routed. If the ID and password are not valid on the specified server, the command execution will fail.
- The administrator must have the appropriate authority to execute the administrative commands on the specified server.

The administrative ID and password used to issue the administrative command are also used on the server where the command is routed. The password is encrypted before it is sent across the network.

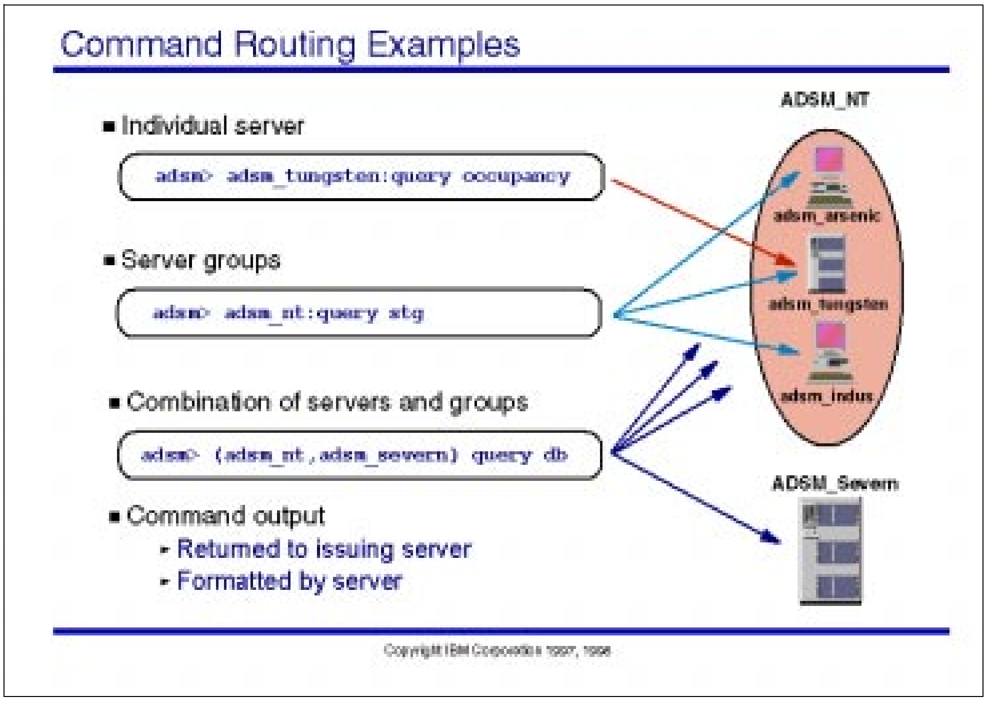
6.2.1.2 New Command Syntax

The new administrative command syntax is of the form:

```
adsm> server name, server_group: administrative command
adsm> (servername, server_group) administrative command
```

The routing information is a prefix to the administrative command. It consists of one or more server or server group names, separated by commas. The end of the routing information is denoted by a colon followed by the administrative command. An alternative syntax is to enclose the server or server group names within parentheses, without the colon before the administrative command. This alternative syntax must be used if command routing is used within server scripts. Wildcards cannot be used within routing information.

If no routing information is not specified, the command is issued on the local server where it was entered.



Command Routing Examples

- Individual server

```
adsm> adsm_tungsten:query occupancy
```

- Server groups

```
adsm> adsm_nt:query stg
```

- Combination of servers and groups

```
adsm> (adsm_nt,adsm_severn) query db
```

- Command output
  - Returned to issuing server
  - Formatted by server

Copyright IBM Corporation 1997, 1998

6.2.2 Command Routing Examples

Administrative commands can be routed to individual servers, server groups, or any combination of servers and server groups.

6.2.2.1 Individual Servers

In the example, the query occupancy is issued against a single server. If there is no definition for this server, or the server cannot be accessed, this command is not issued against any server, and it fails.

6.2.2.2 Server Groups

In the example, the query stgpool command is issued against the ADSM_NT server group. The command is issued to the individual servers that are members of that server group.

6.2.2.3 Combination of Servers and Groups

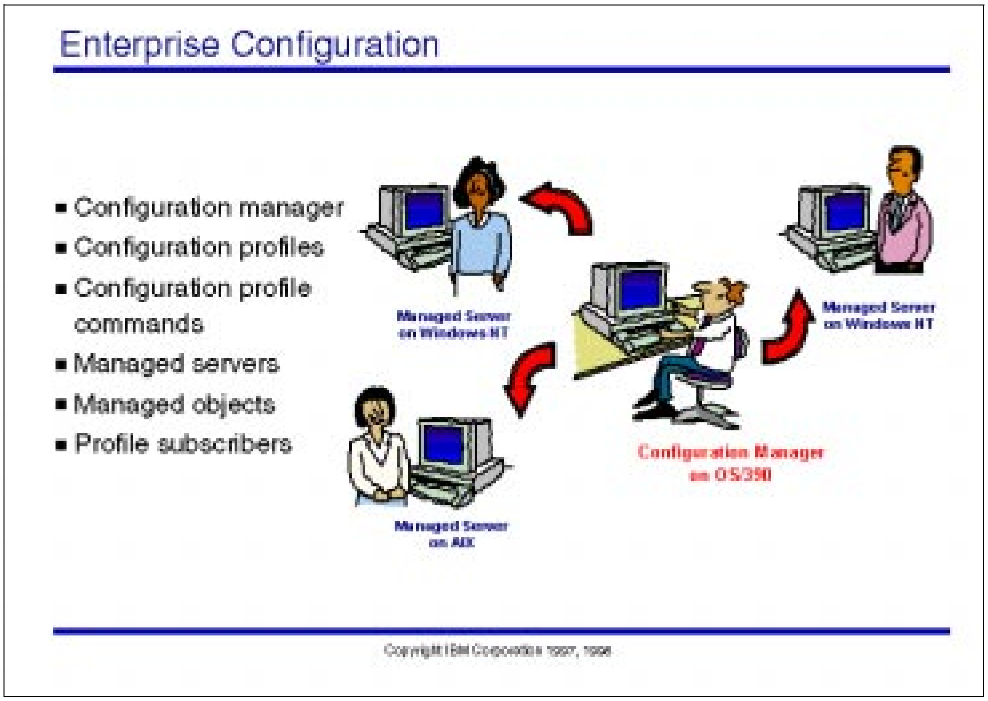
In the example, the query db command is issued against the ADSM_NT server group and the ADSM_SEVERN server. The command is issued against the group members and the individual server.

6.2.2.4 Command Output

The command output from routed commands is returned to the server where the command was issued. The output is formatted so that output from individual servers is contiguous and identifies from which server the output was returned.

Duplicate server names are ignored. If a command is routed to two server groups that contain the same server name, the command is issued only once for that server. If a single server fails to execute a command, a return code is displayed with an associated error message for that server.

6.3 Enterprise Configuration



Enterprise Configuration

- Configuration manager
- Configuration profiles
- Configuration profile commands
- Managed servers
- Managed objects
- Profile subscribers

Managed Server on Windows NT
Managed Server on Windows NT
Configuration Manager on OS/390
Managed Server on AIX

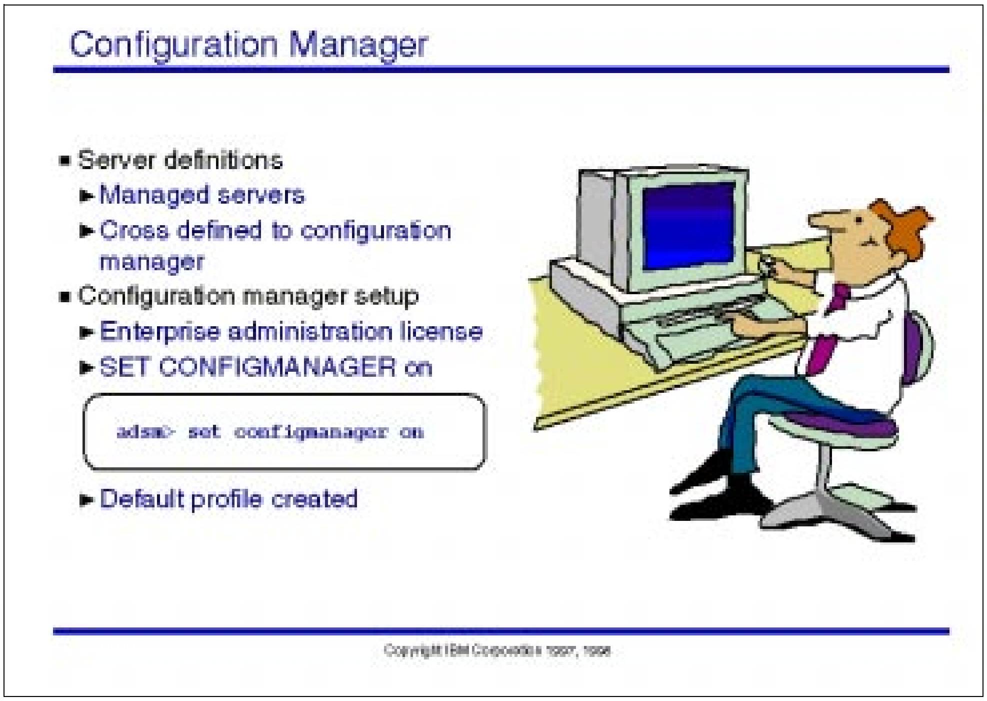
Copyright IBM Corporation 1997, 1998

The ADSM Version 3 enterprise configuration allows an administrator to centrally define configuration information, or *objects*, on a *configuration manager* and then distribute those objects to *managed servers*, using server-to-server communications.

The objects are first defined on the configuration manager by an administrator. The administrator then defines a *configuration profile* and associates the objects with the configuration profile. When the managed server subscribes to this configuration profile, the objects associated with the profile are distributed to the managed server as *managed objects*.

The following topics are covered in this section:

- Configuration manager
- Configuration profiles
- Configuration profile commands
- Managed servers
- Managed objects
- Profile subscribers



Configuration Manager

- Server definitions
  - ► Managed servers
  - ► Cross defined to configuration manager
- Configuration manager setup
  - ► Enterprise administration license
  - ► SET CONFIGMANAGER on

```
adsm> set configmanager on
```

  - ► Default profile created

Copyright IBM Corporation 1997, 1998

6.3.1 Configuration Manager

The configuration manager is the ADSM server used to define a central configuration and propagate it to one or more managed servers. In an enterprise environment there could be one or more configuration managers. However, a managed server can only be subscribed to by one configuration manager at a time.

6.3.1.1 Server Definitions

To propagate the central configuration, the configuration manager must be able to communicate with its managed servers, using server-to-server communications. The managed servers must be defined on the configuration manager with the DEFINE SERVER command. See “Server-to-Server Configuration” on page 180 for a description of the commands used to establish server-to-server communication. The configuration manager must also be defined to the managed servers.

6.3.1.2 Configuration Manager Setup

Central configuration is enabled by a new licensed feature on the ADSM server. The Enterprise Administration license must be registered on the server that will be used as the configuration manager. The REGISTER LICENSE command with the appropriate license key string registers the license.

Once the license has been enabled, the server must be enabled as a configuration manager by issuing the SET CONFIGMANAGER ON command. The QUERY STATUS command can be used to display the values for the CONFIGMANAGER setting. A server cannot be set as a configuration manager if it already subscribed to a configuration profile defined on another configuration manager.

When the SET CONFIGMANAGER ON command is issued, a configuration profile named *default_profile* is created. This default profile contains references to all servers and server group objects defined on the configuration manager.

The CONFIGMANAGER option can be removed by using the SET CONFIGMANAGER OFF command. This command will be successful only if configuration profiles do not exist on the configuration manager.

Configuration Profiles

- Default profile
  - ►Server definitions
  - ►Server group definitions
- Additional profiles
- Managed objects
  - ►Administrators
  - ►Policy domains
  - ►Command schedules
  - ►Server scripts
  - ►Client option sets
  - ►Servers and server groups
- Profile subscribers

Configuration profile : ADSM_RIVERS TUNGSTEN PROFILE

Profile name:

Locked? NO

Description:

Document: Done

Copyright IBM Corporation 1997, 1998

6.3.2 Configuration Profiles

A configuration profile is a set of definitions on the configuration manager that are distributed to managed servers that subscribe to the profile. The configuration profile contains references only to the objects defined on the configuration manager, not the actual objects.

6.3.2.1 Default Profile

The default profile is the first profile created on the configuration manager. It is automatically created when the SET CONFIGMANAGER ON command is issued. The name of the default profile is always default_profile, and it contains associations to all server and server group definitions within the database on the configuration manager.

When a managed server subscribes to a configuration profile, it is automatically subscribed to the default profile as well. A managed server can also directly subscribe to the default profile by specifying the default profile name when using the DEFINE SUBSCRIPTION command.

The default profile can be updated with associations to additional objects for distribution to managed servers. Any additional objects associated with the default profile are distributed to the managed server.

The default profile can be deleted (for example, to avoid the distribution of the server and server group definitions to the managed servers). If the default profile is deleted, it can be re-created by defining a new profile with the same *default_profile* name. The re-created default profile will not contain the server and server group associations. These will have to be reassociated by using the DEFINE PROFASSOCIATION command.

6.3.2.2 Additional Profiles

In addition to the default profile, other configuration profiles can be defined. For example, a configuration profile could be defined for distributing administrator definitions. The administrators, which have to first be defined on the configuration manager, are then associated with this configuration profile.

When a managed server subscribes to this configuration profile, the administrator definitions associated with it are distributed to the managed server. These administrator definitions become managed objects on the managed server.

6.3.2.3 Managed Objects

Objects associated with configuration profiles become managed objects on the managed servers following their distribution. They are managed objects because their attributes are determined on the configuration manager where they were originally defined, and they cannot be updated or modified on the managed server. Managed objects are stored in the managed server database and can be used on the managed server, without a connection to the configuration manager.

The following managed objects can be received from the configuration manager:

- Administrators
- Policy domains
- Administrative command schedules
- Server scripts
- Client option sets
- Servers and server groups

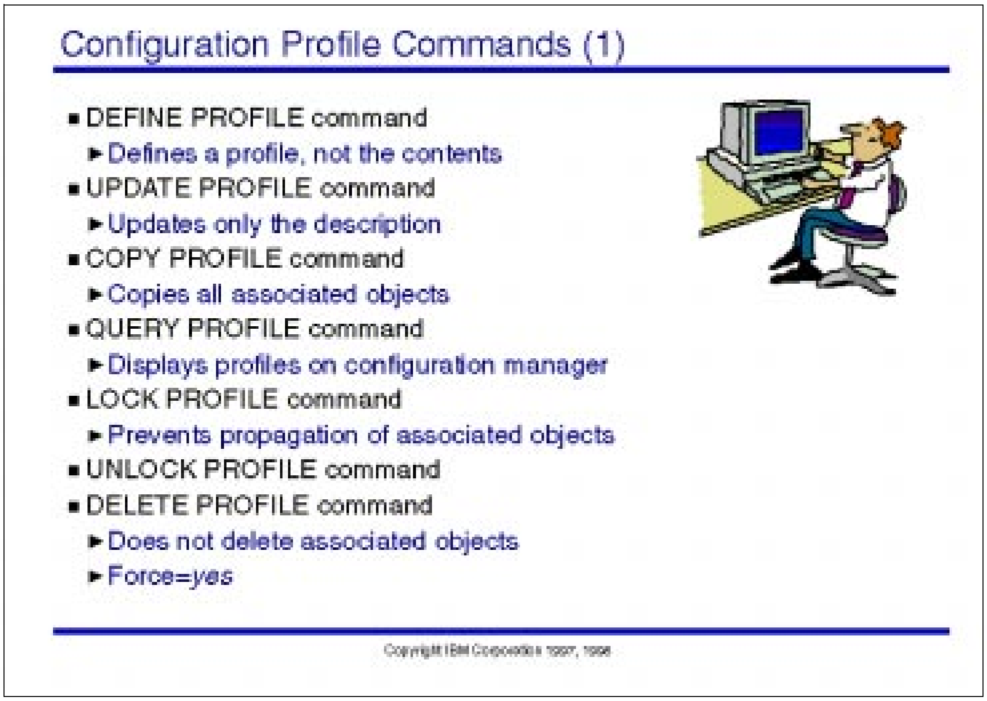
A managed object cannot be modified or deleted on the managed server. Whenever a managed object has to be modified, the modifications must be performed on the configuration manager. The next time the managed server refreshes the associated configuration profile, the modified object will be propagated to the managed server, replacing the previous managed object. Managed servers refresh the configuration profiles they are subscribed to on an administrator-defined refresh interval. If required, the configuration manager can notify the managed server to immediately refresh its configuration.

Managed Objects

Individual objects can be associated with multiple configuration profiles, allowing flexibility in defining central configurations.

6.3.2.4 Profile Subscribers

All managed servers that subscribe to at least one configuration profile on the configuration manager become profile subscribers.



Configuration Profile Commands (1)

- DEFINE PROFILE command
  - Defines a profile, not the contents
- UPDATE PROFILE command
  - Updates only the description
- COPY PROFILE command
  - Copies all associated objects
- QUERY PROFILE command
  - Displays profiles on configuration manager
- LOCK PROFILE command
  - Prevents propagation of associated objects
- UNLOCK PROFILE command
- DELETE PROFILE command
  - Does not delete associated objects
  - Force=*yes*

Copyright IBM Corporation 1997, 1998

6.3.3 Configuration Profile Commands (1)

ADSM central configuration introduces new administrative commands for creating and managing profiles on a configuration manager.

6.3.3.1 DEFINE PROFILE Command

Creates a configuration profile on the configuration manager. Only the configuration profile definition is created. The defined profile does not contain any objects associations:

```
adsm> Define profile tungsten_profile
description=’Configuration profile for server Tungsten’
```

6.3.3.2 UPDATE PROFILE Command

Updates the description field of a configuration profile:

```
adsm> Update profile tungsten_profile
description=’New configuration profile for server Tungsten
```

6.3.3.3 COPY PROFILE Command

Copies a configuration profile and its associated objects to another configuration profile:

```
adsm> Copy profile tungsten_profile arsenic_profile
```

6.3.3.4 QUERY PROFILE Command

Displays all profiles defined on the configuration manager:

```
adsm> Query profile

Configuration       Profile name        Locked?
manager
---------------     ---------------     -------
ADSM_SEVERN         ADMIN                 No
ADSM_SEVERN         ARSENIC               No
ADSM_SEVERN         DEFAULT_PROFILE       No
ADSM_SEVERN         NT_ADMINS             No
ADSM_SEVERN         TUNGSTEN_PROFI-       No
                     LE
```

The QUERY PROFILE command can also be used on managed servers to display profiles defined on the configuration manager.

6.3.3.5 LOCK PROFILE Command

Locks a configuration profile so that the associated objects are not propagated to subscribing managed servers:

```
adsm> Lock profile tungsten_profile 45
```

The optional parameter is the time, in minutes, that the profile will remain locked. After that period it is unlocked. A value of 0 locks the profile permanently.

6.3.3.6 UNLOCK PROFILE Command

Unlocks a previously locked configuration profile, allowing the associated objects to be propagated to subscribing managed servers:

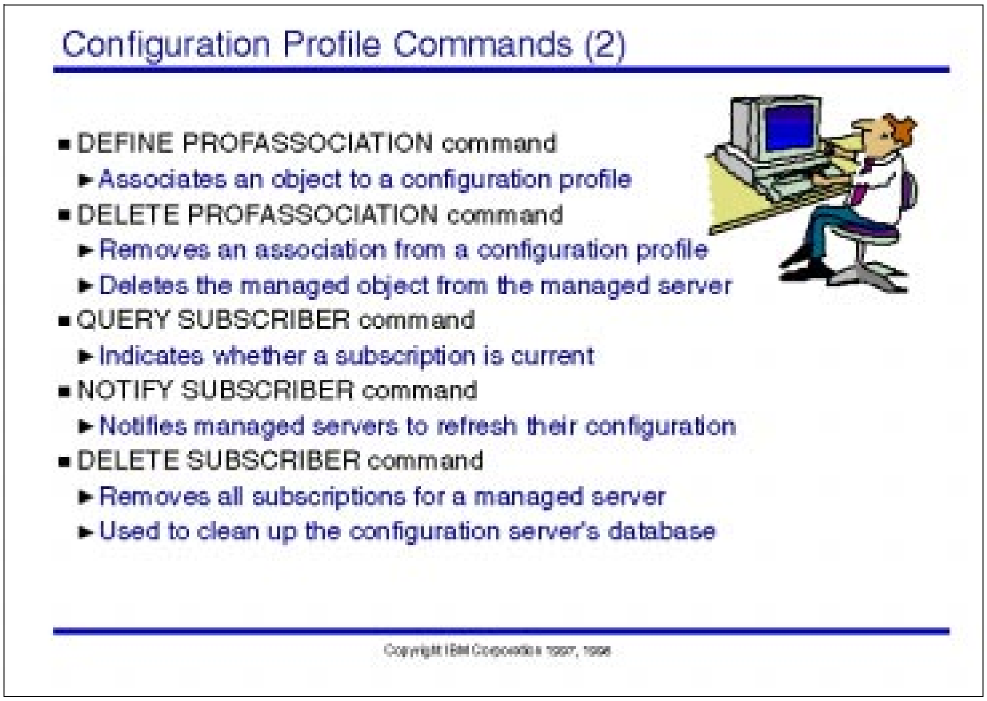
```
adsm> Unlock profile tungsten_profile
```

6.3.3.7 DELETE PROFILE Command

Deletes a profile and its references to any associated objects. It does not delete the objects associated with the profile from the configuration manager or the managed objects from the managed servers:

```
adsm> Delete profile tungsten_profile
```

Managed servers should remove their subscription to the profile before it is deleted. If this is not done, the managed server will continue attempting to refresh the profile, even though it has been deleted. If managed servers still have subscriptions to the profile, a FORCE=YES parameter can be specified to force the profile deletion. If one or more managed servers still have a subscription to that profile, the FORCE=YES parameter must be specified to delete the profile.



Configuration Profile Commands (2)

- DEFINE PROFASSOCIATION command
  - ► Associates an object to a configuration profile
- DELETE PROFASSOCIATION command
  - ► Removes an association from a configuration profile
  - ► Deletes the managed object from the managed server
- QUERY SUBSCRIBER command
  - ► Indicates whether a subscription is current
- NOTIFY SUBSCRIBER command
  - ► Notifies managed servers to refresh their configuration
- DELETE SUBSCRIBER command
  - ► Removes all subscriptions for a managed server
  - ► Used to clean up the configuration server's database

Copyright IBM Corporation 1997, 1998

6.3.4 Configuration Profile Commands (2)

In addition to the commands for creating profiles, there are new commands for profile associations and profile subscribers.

6.3.4.1 DEFINE PROFASSOCIATION Command

Associates one or more objects with a configuration profile to distribute them to subscribing managed servers. Multiple object types can be associated with a profile, and a *match all* definition can be used to associate all defined objects of a type:

```
adsm> Define profassociation tungsten_profile

admins=*

adscheds=full_db_backup,del_volume_history
```

This example associates all administrator IDs defined on the configuration manager, and two administrative schedules, with the tungsten_profile. Using an asterisk (*) simplifies definitions where all objects of a particular type on the configuration manager are to be propagated to managed servers. In this example, as administrator IDs are added on the configuration manager they are automatically propagated to the subscribing managed servers. If an administrator is then removed on the configuration manager, the corresponding administrator managed object will also be removed on the managed servers when the profile is next refreshed.

The objects must be defined on the configuration manager before they can be associated with a configuration profile.

6.3.4.2 DELETE PROFASSOCIATION Command

Deletes the association of one or more objects from a specific configuration profile. When the profile association is deleted, the objects are no longer distributed to the subscribing managed server:

```
adsm> Delete profassociation tungsten_profile

adscheds=del_volume_history
```

This example deletes the previously defined profile association for the *del_volume_history* administrative schedule. When the managed server next refreshes the configuration profile, the configuration manager notifies it of the objects' deletion. The managed server automatically deletes these objects unless they are associated with another profile to which the managed server subscribes.

6.3.4.3 QUERY SUBSCRIBER Command

Shows information about subscribers and their subscriptions to configuration profiles. The information includes the name of the configuration profile, the name of the subscribing managed server, whether the configuration information is current, and when it was last updated:

```
adsm> Query subscriber
Subscriber         Profile name       Is current?         Last update
                                                           date/time
---------------    ---------------    ------------     --------------------
ADSM_ARSENIC       DEFAULT_PROFILE         No           07/10/98 16:06:46
ADSM_INDUS         DEFAULT_PROFILE         No           07/14/98 14:17:47
ADSM_TUNGSTEN      DEFAULT_PROFILE        Yes           07/14/98 16:09:50
ADSM_TUNGSTEN      TUNGSTEN_PROFI-        Yes           07/14/98 16:09:50
                    LE
```

6.3.4.4 NOTIFY SUBSCRIBER Command

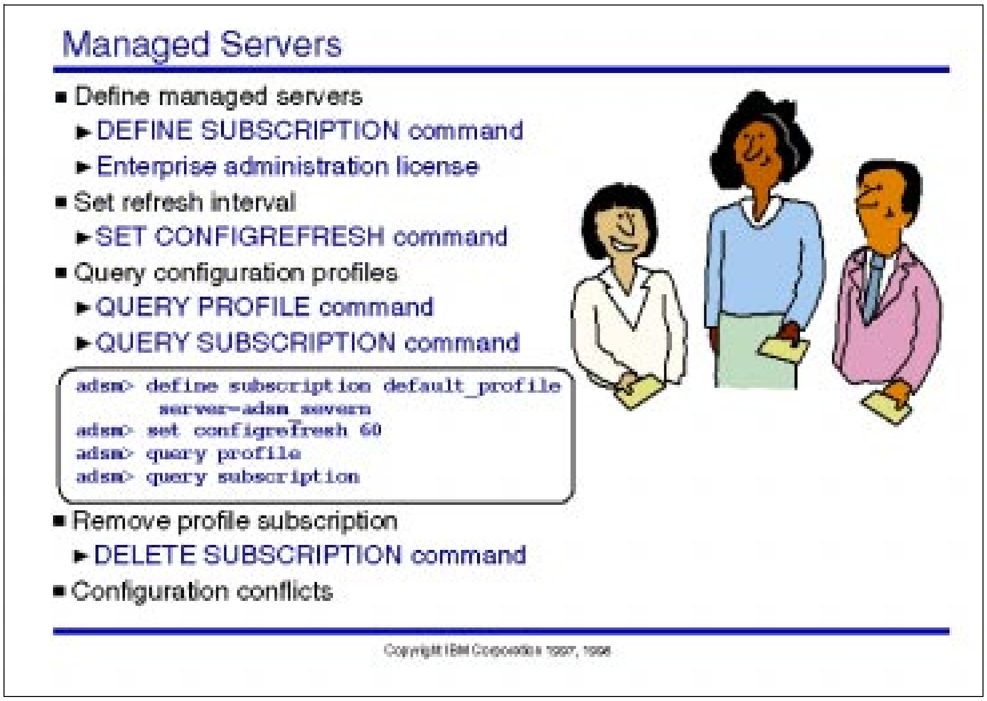
Notifies the subscribing managed servers that a configuration profile should be refreshed immediately from the configuration manager. When this command is issued, the managed servers refresh the configuration profile specified:

```
adsm> Notify subscriber profile=tungsten_profile
```

6.3.4.5 DELETE SUBSCRIBER Command

Deletes all subscriptions for a managed server from the configuration manager database. This command has to be used to delete obsolete managed server subscriptions in cases where the managed server no longer exists or is unable to notify the configuration manager after deleting a subscription:

```
adsm> Delete subscriber adsm_tungsten
```



Managed Servers

- Define managed servers
  - DEFINE SUBSCRIPTION command
  - Enterprise administration license
- Set refresh interval
  - SET CONFIGREFRESH command
- Query configuration profiles
  - QUERY PROFILE command
  - QUERY SUBSCRIPTION command

```
adsm> define subscription default_profile
          server=adsm_severn
adsm> set configrefresh 60
adsm> query profile
adsm> query subscription
```

- Remove profile subscription
  - DELETE SUBSCRIPTION command
- Configuration conflicts

Copyright IBM Corporation 1997, 1998

6.3.5 Managed Servers

A managed server is a server that is subscribed to one or more configuration profiles on a configuration manager. A managed server receives objects on the configuration manager that are associated with the configuration profile. A managed server can only subscribe to one configuration manager, and it cannot also be a configuration manager to other managed servers.

6.3.5.1 Define Managed Servers

A server becomes managed by subscribing to one or more configuration profiles on a configuration manager. This is done by using the DEFINE SUBSCRIPTION command specifying the profile name to subscribe to and the configuration manager name:

```
adsm> define subscription default_profile server=adsm_severn
```

This example illustrates a server subscribing to the default profile on the adsm_servern configuration manager. The first subscription made by a server does not have to be to the default profile. If the subscription is to another profile, the server is also automatically subscribed to the default profile on the configuration manager.

To use the DEFINE SUBSCRIPTION command, the *Enterprise Administration* license must be registered on the managed server through the REGISTER LICENSE command. The server must also have a valid server definition to the configuration manager.

6.3.5.2 Set Refresh Interval

When a managed server subscribes to a configuration profile, it periodically contacts the configuration manager to refresh the subscribed configuration profiles for any updates that have been made on the configuration manager. The configuration refresh interval is set on the managed server with the SET CONFIGREFRESH command and specifies the time in minutes that elapses between attempts by the managed server to contact the configuration manager for configuration refreshes. The QUERY STATUS command can be used to display the value for the CONFIGREFRESH option.

6.3.5.3 Query Configuration Profiles

Managed servers can query the configuration manager for all defined profiles and those profiles to which it is subscribed.

The QUERY PROFILE command is used on a managed server to query profiles defined on the configuration manager:

```
adsm> query profile uselocal=no
Configuration       Profile name        Locked?
manager
---------------     ---------------     -------
ADSM_SEVERN         ADMIN                 No
ADSM_SEVERN         ARSENIC               No
ADSM_SEVERN         NT_ADMINS             No
ADSM_SEVERN         TUNGSTEN_PROFI-       No
                     LE
```

By default, profiles known locally to the managed server are queried. Profile names are updated in the managed server database every time the managed server refreshes its configuration from the configuration manager. The QUERY PROFILE *uselocal=no* option specifies that the managed server should contact the configuration manager for the list of profiles defined on that server.

The QUERY SUBSCRIPTION command is used on a managed server to query profiles on the configuration manager to which it is subscribed:

```
adsm> query subscription

Configuration       Profile name            Last update
manager                                      date/time
---------------     ---------------     --------------------
ADSM_SEVERN         DEFAULT_PROFILE
ADSM_SEVERN         NT_ADMINS           07/10/1998 16:08:17
```

6.3.5.4 Remove Profile Subscription

The DELETE SUBSCRIPTION command can be used on managed servers to remove the subscription of that server from the specified profile on the configuration manager:

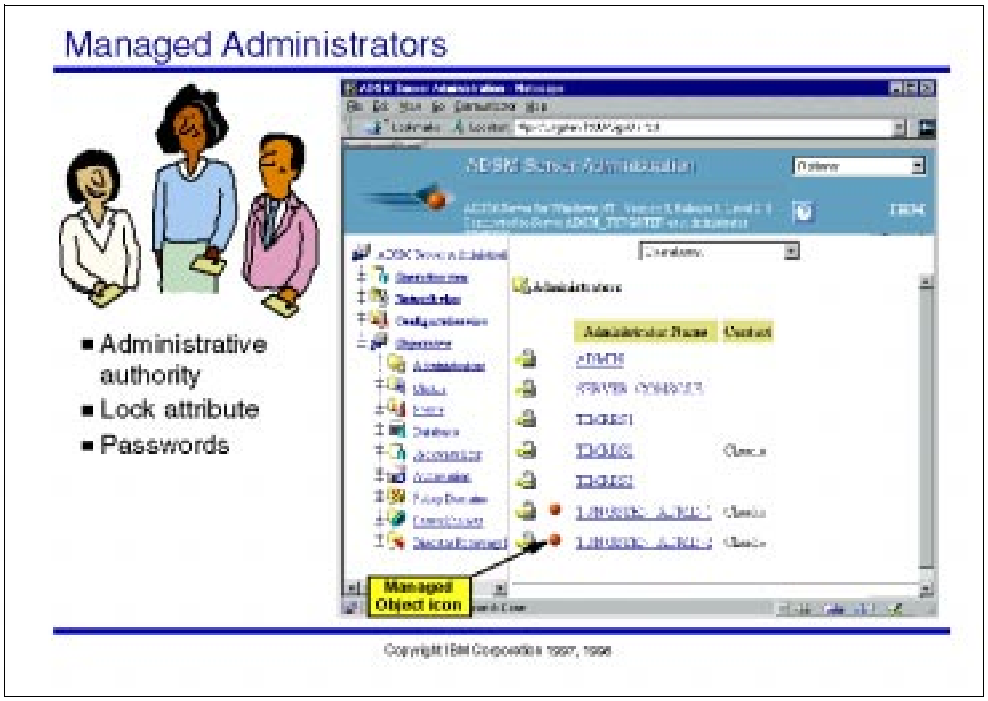
```
adsm> delete subscription nt_admins
```

The command deletes a configuration profile subscription. Managed objects associated with the profile are left on the managed server. These objects are no longer managed and can be modified locally on the managed server.

If the managed objects are to be deleted, the *discardobjects=yes* option can be specified. This option deletes the managed object associated with the subscription being deleted.

6.3.5.5 Configuration Conflicts

When a managed server receives or refreshes a configuration profile, all objects associated with that profile replace any locally defined objects of the same type on the managed server with the same name. For example, if an administrator ID is locally defined on the managed server and an administrator ID with the same name is also defined and associated with the configuration profile on the configuration manager, the local definitions are replaced by the definitions obtained from the configuration manager.



Managed Administrators

- Administrative authority
- Lock attribute
- Passwords

Copyright IBM Corporation 1997, 1998

6.3.6 Managed Administrators

Any administrator ID defined on the configuration manager can be added to a configuration profile with the exception of the SERVER_CONSOLE ID. These administrator IDs are propagated to subscribing managed servers as managed objects. If such an ID is subsequently deleted on the configuration manager, the associated managed object on the managed server is also deleted. Exceptions to this are: If the ID is in use on the managed server it will not be deleted, and an administrator with system privilege will not be deleted if such deletion would leave the managed server without a system administrator.

6.3.6.1 Administrative Authority

Whenever an administrator ID is propagated, all privileges of that administrator ID on the configuration manager are propagated with the administrator ID to the managed servers. Therefore, a managed administrator ID always has the same authority on the configuration manager and on all managed servers to which the ID has been propagated.

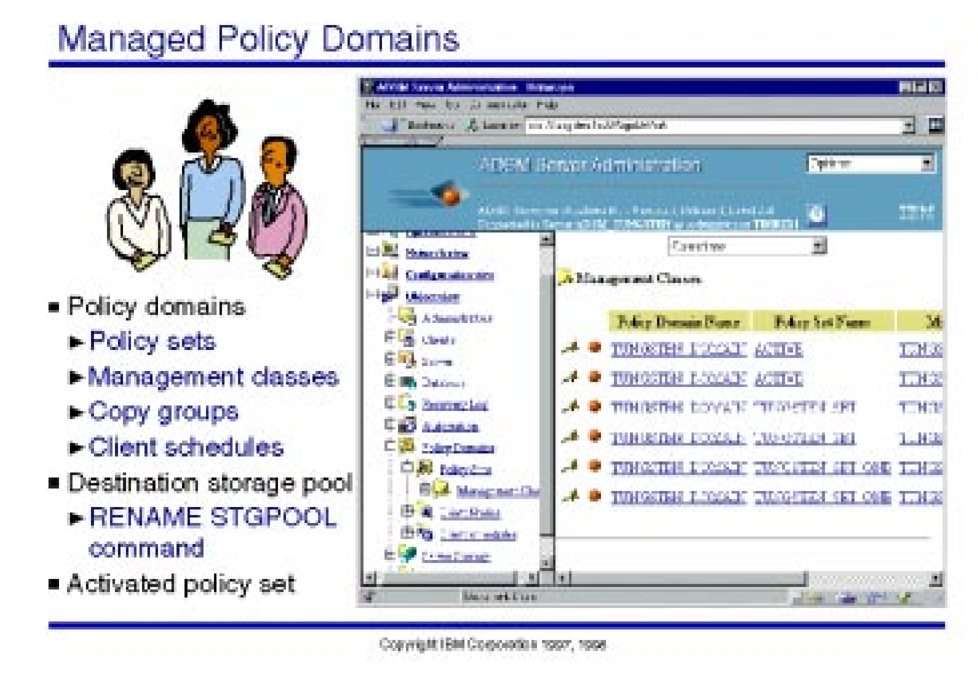
6.3.6.2 Lock Attribute

The lock attribute of an administrator ID on the configuration manager is not propagated to the managed servers. This is a useful security feature. Administrator IDs can be defined for use only on managed servers by locking them on the configuration manager. When the ID is propagated, it will remain locked on the configuration manager, but it will not be locked on the subscribing managed servers. Such an ID can be used to administer the managed server but could not be used to log in to, or route commands to, the configuration manager.

6.3.6.3 Passwords

The password of an administrator ID is propagated to the managed servers. It is not possible to update the password of a managed administrator ID on the managed servers. The password must be updated on the configuration manager, and the managed servers must be notified to refresh the configuration profile.

If the *password expiration* value is not the same on the configuration manager and the managed servers, the password of a managed administrator ID can be changed if it is expired. As an example, the *password expiration* value is set to 90 on the configuration manager and to 10 on the managed server. If a managed administrator ID logs in to the managed server and his password has expired, it can be changed. This results in a password on the configuration manager that is different from the password on the managed server for the same administrator ID, until the password is next changed on the configuration manager, and the managed server is notified to refresh the configuration profile.



Managed Policy Domains

- Policy domains
  - Policy sets
  - Management classes
  - Copy groups
  - Client schedules
- Destination storage pool
  - RENAME STGPOOL command
- Activated policy set

Copyright IBM Corporation 1997, 1998

6.3.7 Managed Policy Domains

Policy domains can be associated with a configuration profile. All objects within the policy domain are propagated as managed objects to the managed servers.

6.3.7.1 Policy Domains

A policy domain defined on the configuration manager can be associated with a configuration profile by associating the policy domain name. When such a profile is subscribed to by a managed server, the policy domain and its associated objects (policy set, management classes, and copy groups) will be propagated to the managed server and become managed objects on that server. If a locally defined policy domain with the same name exists, it will be overwritten. If that locally defined domain has nodes assigned to it, the propagation will fail. The nodes must be reassigned to another domain before the propagation can complete.

Client schedules defined for the policy domain on the configuration manager are also propagated to the managed server as an entity of the policy domain structure. The propagated client schedules on the managed server are managed objects and cannot be modified or deleted. The only operation allowed on a managed client schedule is its association with one or more client nodes.

After a policy domain is associated with and propagated to a configuration profile, any modifications made on the configuration manager are also propagated when the managed server next refreshes the profile.

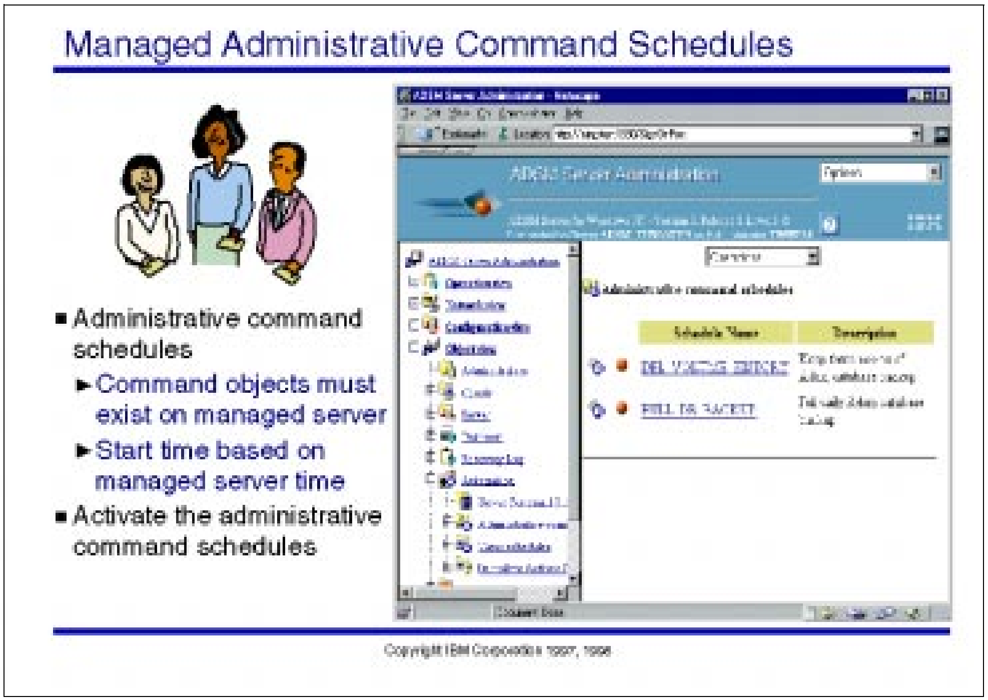
When a policy domain associated with a configuration profile is deleted on the configuration manager, the managed object on the manager server is also deleted. Any nodes assigned to that domain must be removed before the deletion can complete.

6.3.7.2 Destination Storage Pool

The copy groups that are propagated with the profile contain pointers to destination storage pool names. These copy groups are managed objects on the managed server, and their destination storage pool name cannot be modified. To handle situations where the destination storage pool name in the propagated copy group does not match the actual storage pool name on the managed server, there is a new RENAME STGPOOL command. This command can be used on the managed server to rename an existing storage pool to match the name defined in the copy group managed object.

6.3.7.3 Activated Policy Set

The policy domain on the configuration server to be distributed can contain an activated policy set. However, the inactive policy set is propagated to the managed servers. After the inactive policy set has been propagated, it must be validated and activated on the managed server.



Managed Administrative Command Schedules

- Administrative command schedules
  - Command objects must exist on managed server
  - Start time based on managed server time
- Activate the administrative command schedules

Copyright IBM Corporation 1997, 1998

6.3.8 Managed Administrative Command Schedules

Administrative command schedules can be associated with a configuration profile.

6.3.8.1 Administrative Command Schedules

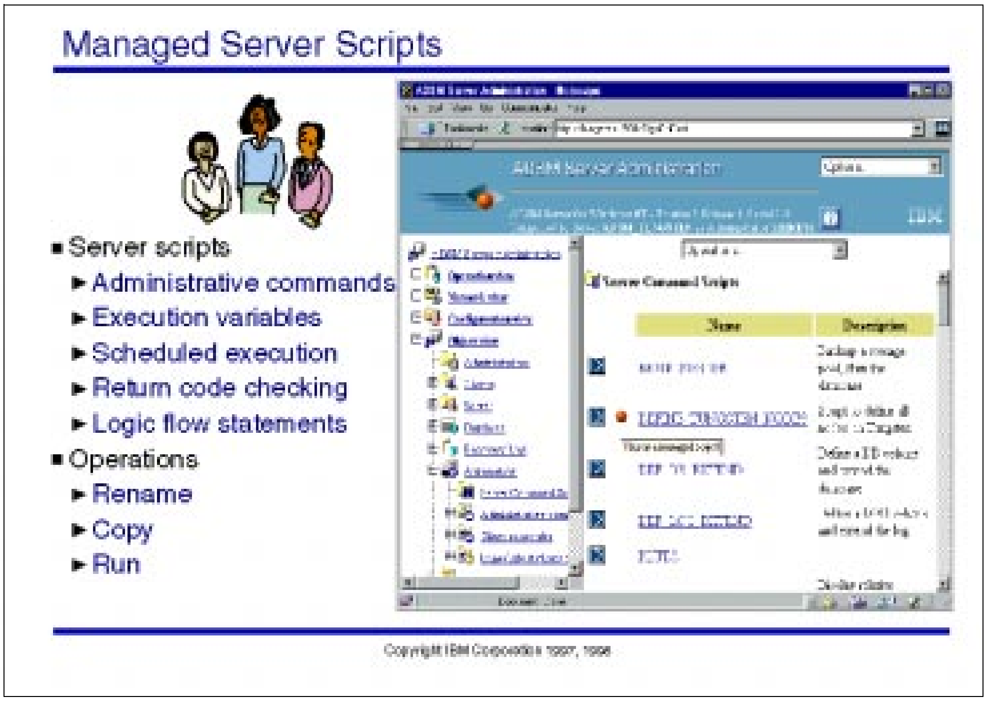
An administrative command schedule is a server schedule that executes administrative commands on an ADSM server. For example, two administrative command schedules could be defined on the configuration manager, the first one backing up the ADSM database, and the second deleting old database backups from the volume history. These administrative command schedules can be associated with a configuration profile to be distributed to managed servers so that the same database backup policy applies to the managed servers.

When defining administrative command schedules to be propagated, care should be taken in specifying commands or scripts to be executed. All specified commands or scripts must exist on the managed server where the schedule will execute.

6.3.8.2 Activate the Administrative Command Schedules

An administrative command schedule must be defined on the configuration manager before it can be associated with a configuration profile. It can be defined either as an active or inactive administrative command schedule.

After propagation, the managed administrative command schedules are not activated on the managed server, even if they were on the configuration manager. The schedule must be activated on the managed servers in order for it to be executed. Such activated managed schedules are not automatically deleted on the managed server during a refresh following their deletion on the configuration manager. They must first be deactivated before they can be deleted.



Managed Server Scripts

- Server scripts
  - Administrative commands
  - Execution variables
  - Scheduled execution
  - Return code checking
  - Logic flow statements
- Operations
  - Rename
  - Copy
  - Run

Copyright IBM Corporation 1997, 1998

6.3.9 Managed Server Scripts

Server scripts defined on the configuration manager can be associated with a configuration profile for propagation to managed servers.

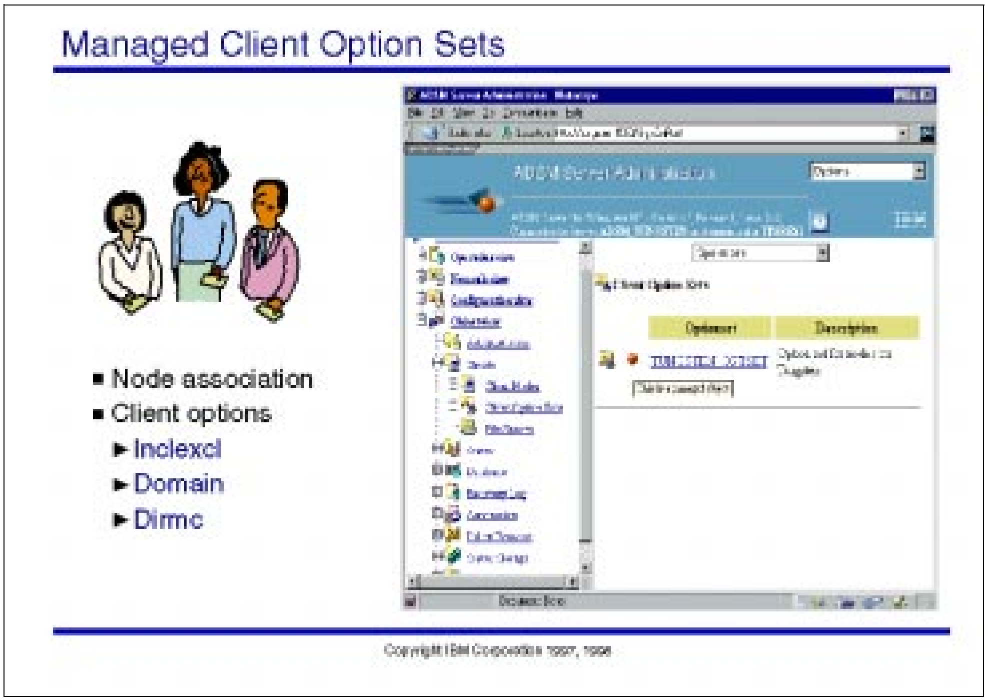
6.3.9.1 Server Scripts

A server script is a set of administrative commands that are sequentially executed. The scripts use variables and logic flow statements. A simple server script to be distributed could contain all administrative commands required to define all ADSM clients to an ADSM managed server.

Server script execution can be scheduled as any other administrative command by using the RUN SCRIPT command.

6.3.9.2 Operations

As with other managed objects, managed server scripts cannot be updated, renamed, or deleted on the managed servers. All those operations have to be performed on the configuration manager, and the managed servers have to be notified to update their configuration. However, a managed server script can be copied on the managed servers. The copied server scripts are unmanaged and can be executed on the managed servers.



Managed Client Option Sets

- Node association
- Client options
  - Inclexcl
  - Domain
  - Dirmc

Copyright IBM Corporation 1997, 1998

6.3.10 Managed Client Option Sets

Client option sets defined on the configuration manager can be associated with a configuration profile for propagation to managed servers. The client option set name is associated with the configuration profile. However, all client options defined in the client option set are propagated to the managed servers.

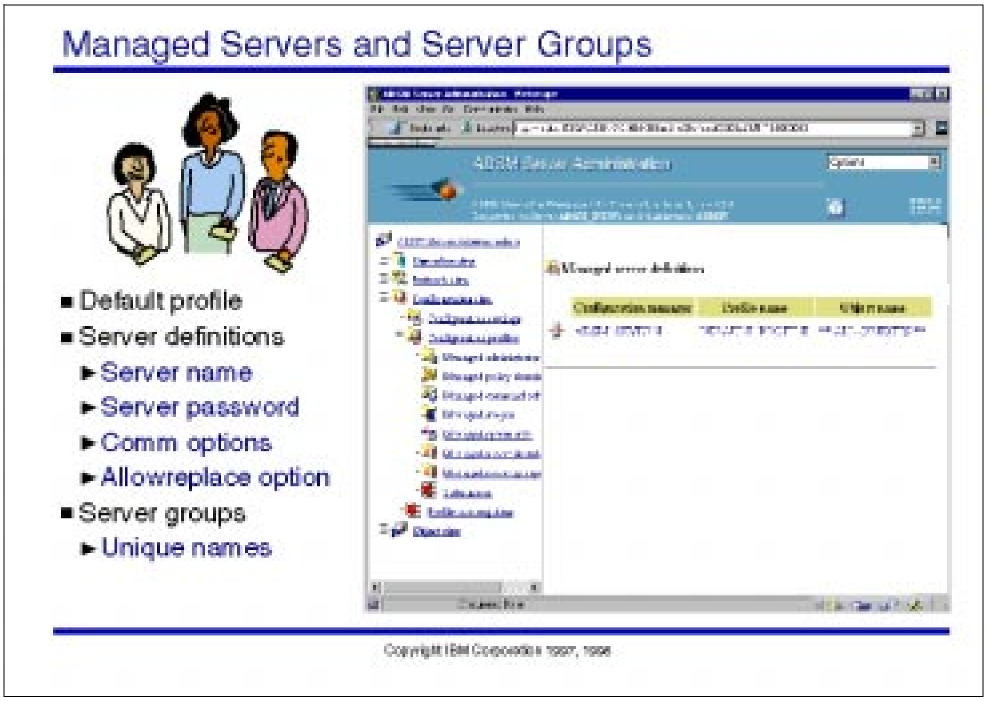
6.3.10.1 Node Association

The association of client nodes with client option sets is not part of the client option set and is not propagated to the managed servers. This node association must be performed on the managed server by using the REGISTER NODE or UPDATE NODE command with the *cloptset=* option.

6.3.10.2 Client Options

All client options within a managed client option set are managed objects and can neither be modified nor deleted on a manager server.

Special attention must be paid for those client options that refer to objects like drives or filesystems (Domain option), paths (Inclexcl option), or management classes; All such objects must exist on the managed servers; it does not matter whether they exist or not on the configuration manager.



Managed Servers and Server Groups

- Default profile
- Server definitions
  - Server name
  - Server password
  - Comm options
  - Allowreplace option
- Server groups
  - Unique names

Copyright IBM Corporation 1997, 1998

6.3.11 Managed Servers and Server Groups

Server and server group definitions defined on the configuration manager can be associated with a configuration profile for propagation to managed servers.

6.3.11.1 Default Profile

When the SET CONFIGMANAGER ON command is issued on a server to create a configuration manager, a default profile is automatically created on that server. This default profile has all server and server group definitions on the configuration manager associated with it. When a managed server subscribes to a configuration profile, it is automatically subscribed to the default profile as well. In addition to the default profile, server and server group definitions can be associated with other configuration profiles.

6.3.11.2 Servers Definitions

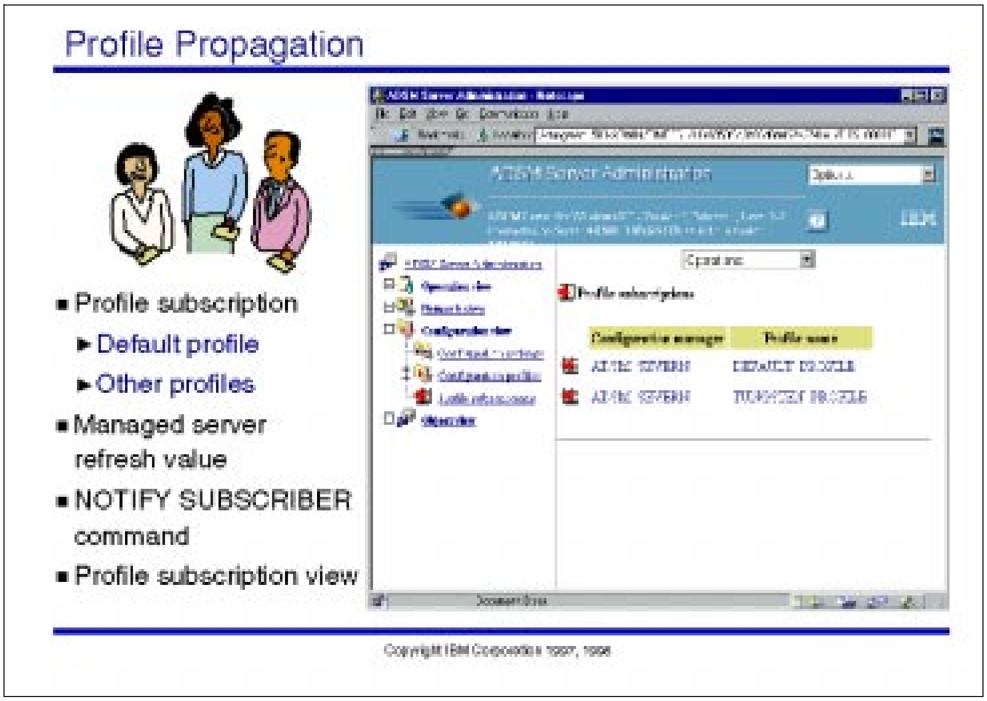
When a server is associated with a configuration profile, only a subset of the server definitions is propagated to a managed server. These are the communication options required to establish a server-to-server connection: server name, communications method, TCP/IP address, server password, URL, and description. The propagated server definition becomes a managed object on the managed server and cannot be modified.

A server definition propagated from a configuration manager is different from other managed objects in that is does not automatically replace a locally defined server definition with the same name. When a server is defined with the DEFINE SERVER command, there is an option, *allowreplace=yes/no*, that determines whether a locally defined server on a managed server can be replaced by a server definition of the same name, propagated from the configuration manager. The default is no.

6.3.11.3 Server Groups

Server groups defined on the configuration manager can be associated with a configuration profile and propagated to a subscribing managed server.

A server group propagated from a configuration manager replaces a locally defined server group with the same name on the managed server. However, servers and server groups must have unique names. If a server group is propagated and there is a *server definition* with the same name on the managed server, the server group propagation will fail.



Profile Propagation

- Profile subscription
  - ► Default profile
  - ► Other profiles
- Managed server refresh value
- NOTIFY SUBSCRIBER command
- Profile subscription view

Copyright IBM Corporation 1997, 1998

6.3.12 Profile Propagation

There are three ways in which configuration profiles are propagated from the configuration manager to managed servers. The refresh operation is always performed by the managed server.

6.3.12.1 Profile Subscription

When a managed server subscribes to its first configuration profile, it is also subscribed to the default profile of its configuration manager. This process creates two subscriptions on the managed server and immediately propagates the objects associated with both profiles to the managed server.

6.3.12.2 Managed Server Refresh Value

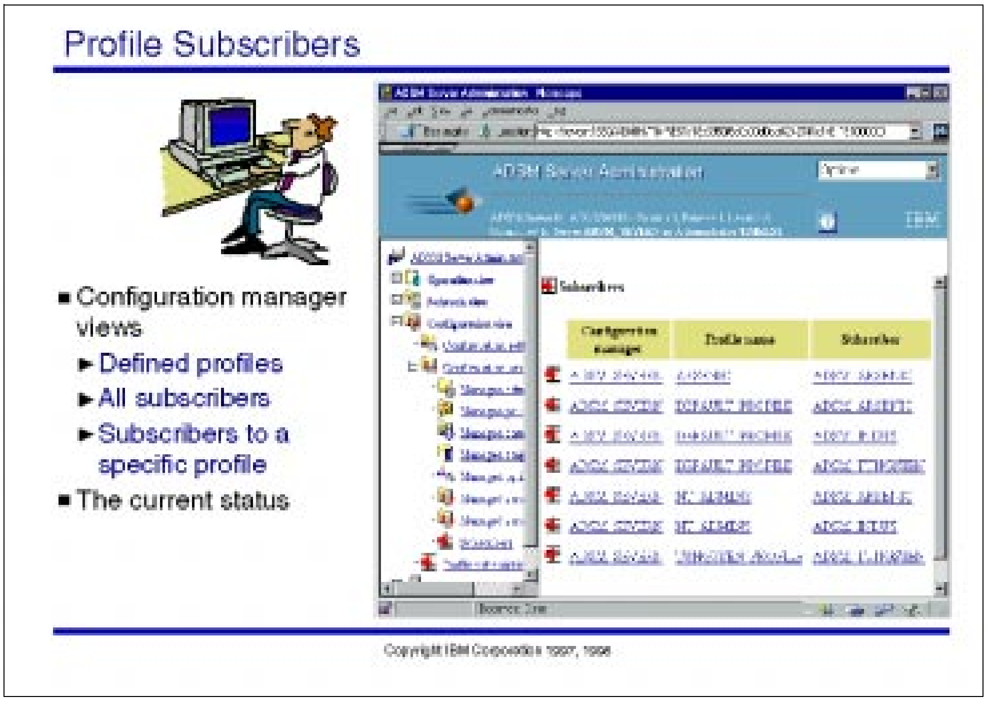
The value set by the SET CONFIGREFRESH command on the managed server determines the interval, in minutes, that elapses before the managed server contacts the configuration manager to refresh profiles to which it is subscribed. When the SET CONFIGREFRESH command is issued with a value greater than zero, the managed server immediately contacts the configuration manager to refresh the configuration. Once the refresh is complete, the timer starts counting down to the next refresh, according to the value specified.

6.3.12.3 NOTIFY SUBSCRIBER Command

The configuration manager cannot *push* profiles to managed servers.The NOTIFY SUBSCRIBER command can be issued on the configuration manager to notify one or more managed servers to immediately refresh their subscriptions from the configuration manager.

6.3.12.4 Profile Subscription View

A managed server administrator uses the QUERY SUBSCRIPTION command to view subscriptions for that server. The command can be issued only on managed servers or the Web administrative interface. The administrator can see only current subscriptions for that managed server.



Profile Subscribers

- Configuration manager views
  - Defined profiles
  - All subscribers
  - Subscribers to a specific profile
- The current status

Copyright IBM Corporation 1997, 1998

6.3.13 Profile Subscribers

Whenever a managed server subscribes to a configuration profile, it becomes a subscriber on the configuration manager. An administrator on the configuration manager can view all profile subscribers.

6.3.13.1 Configuration Manager Views

The configuration manager administrator can use the QUERY SUBSCRIBER command to view all profiles and their subscribers. The command can be issued only on a configuration manager or the Web administrative interface.

Views are also available to display all managed servers that are subscribed to a specific configuration profile, and all profiles to which a specific managed server is subscribed. These views display the current status of the subscription between the configuration and subscribing managed servers.

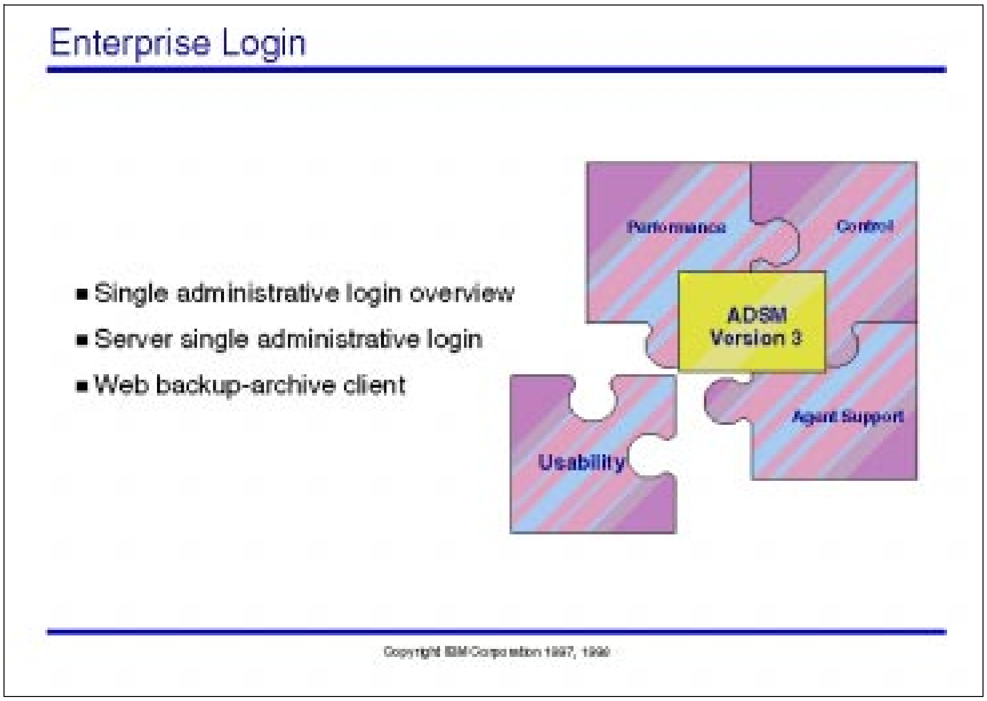
6.3.13.2 The Current Status

The current status is based on the currency of the configuration profile stored on the database of the configuration manager and on the database of the managed servers.

If both the configuration manager and the managed server have the same profile, the current status indicates YES. If the configuration manager profile is newer than that of the managed server, the current status indicates NO.

The current status can also indicate UNCERTAIN; that is, the managed server has a more current profile than that of the configuration manager. This status can occur if the configuration manager database is restored to a previous level or a profile is deleted on the configuration manager while subscriptions remain on managed servers.

Chapter 7. Enterprise Login



Enterprise Login

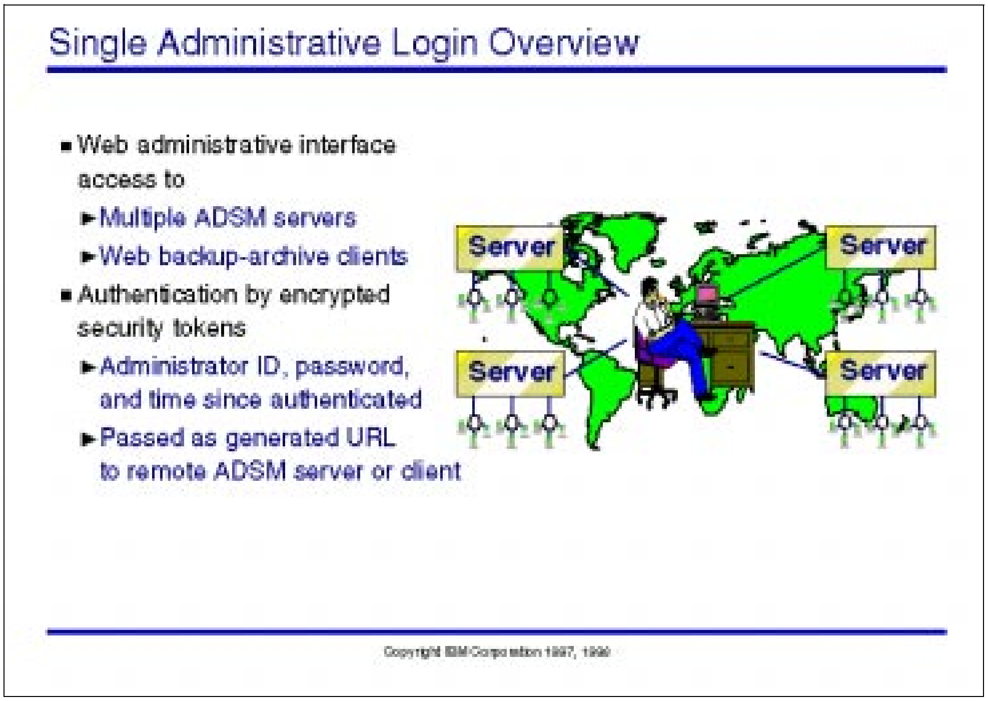
- Single administrative login overview
- Server single administrative login
- Web backup-archive client

Performance
Control
ADSM
Version 3
Agent Support
Usability

This chapter describes the ADSM Version 3 single administrative login features for accessing multiple ADSM servers and clients in an enterprise. The following topics are covered:

- Single administrative login overview
- Server single administrative login
- Web back-archive client

7.1 Single Administrative Login Overview



Single Administrative Login Overview

- Web administrative interface access to
  - Multiple ADSM servers
  - Web backup-archive clients
- Authentication by encrypted security tokens
  - Administrator ID, password, and time since authenticated
  - Passed as generated URL to remote ADSM server or client

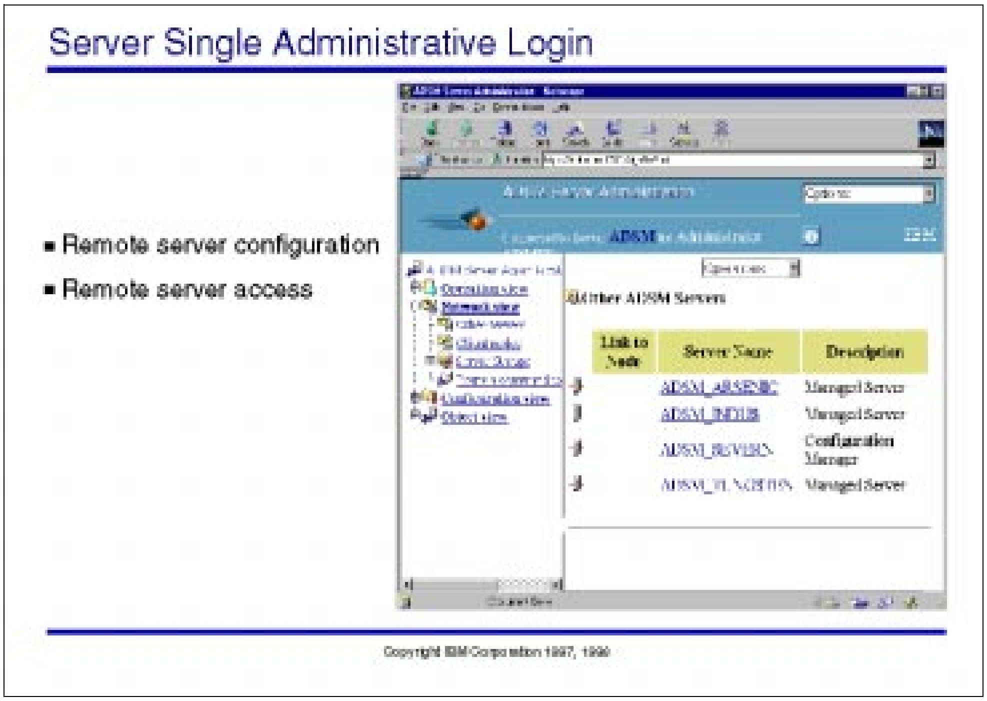
Server
Server
Server
Server

Copyright IBM Corporation 1997, 1998

ADSM Version 3.1.2 introduces new single administrative login functions. Single administrative login enables an administrator to log in to ADSM, using the Web administrative interface, and then access other suitably configured ADSM systems. Administrators can navigate in a seamless manner from one server to another without having to reauthenticate their administrator userids on each server. In addition, administrators and helpdesk personnel with the appropriate authority can perform remote client operations, using the new Web backup-archive client interface.

Administrators are required to authenticate when they first log in to an ADSM server. Once authenticated, their security credentials (administrator userid, password, and elapsed time since they authenticated) are used to authenticate access to other ADSM servers and Web backup-archive clients. Seamless access to servers and Web backup-archive clients is enabled through the creation of hyperlink icons in the Network view of the Web administrative interface (4.1.1, “ADSM Enterprise Console” on page 112). The hyperlink contains the URL for the ADSM servers and Web backup-archive clients to be accessed. When an administrator connects to another server or Web backup-archive by clicking on a hyperlink icon, an extended URL is generated. This extended URL contains the address of the server or client, the userid of the administrator, and an encrypted security token containing details of the administrators security credentials. The server or client being accessed uses the administrator userid and security token to establish that the administrator has authority to access the system. If the administrator is authorized, access is granted. If the administrator is not authorized, or the security token has timed out, a login prompt is displayed requiring the administrator to reauthenticate the userid.

7.2 Server Single Administrative Login



Server Single Administrative Login

- Remote server configuration
- Remote server access

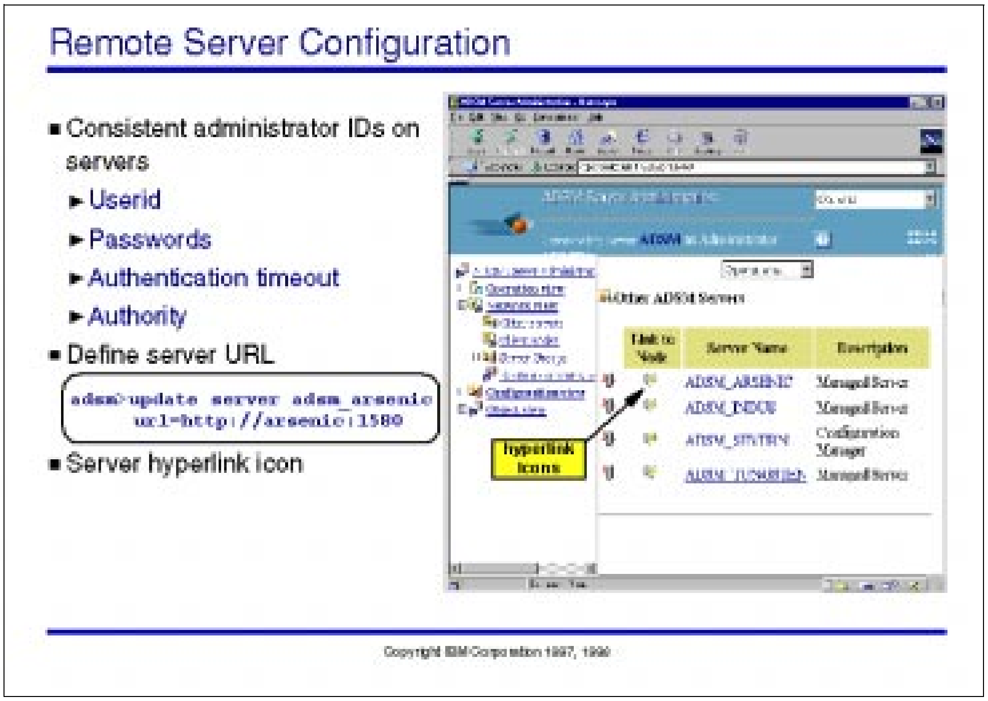
Other ADSM Servers

Link to Node	Server Name	Description
	ADSM_ARSENIC	Managed Server
	ADSM_INDIUM	Managed Server
	ADSM_[illegible]	Configuration Manager
	ADSM_TUNGSTEN	Managed Server

Copyright IBM Corporation 1997, 1999

This section covers configuration and use of single administrative login to access remote ADSM servers. The following topics are covered:

- Remote server configuration
- Remote server access



Remote Server Configuration

- Consistent administrator IDs on servers
  - Userid
  - Passwords
  - Authentication timeout
  - Authority
- Define server URL

```
adsm>update server adsm_arsenic
      url=http://arsenic:1580
```

- Server hyperlink icon

Other ADSM Servers

Link to Web	Server Name	Description
	ADSM_ARSENIC	Managed Server
	ADSM_[illegible]	Managed Server
	ADSM_[illegible]	Configuration Manager
	ADSM_[illegible]	Managed Server

Copyright IBM Corporation 1997, 1999

7.2.1 Remote Server Configuration

Single administrative login is automatically enabled with ADSM Version 3.1.2. However, certain configuration steps must be performed to fully exploit it.

7.2.1.1 Consistent Administrator IDs on Servers

The administrative IDs and passwords to be used must be the same on all servers that will be accessed. The password expiration period for the servers should also be set to the same value through the SET PASSEXP command. If the password expiration is not the same, access may be denied because the password has expired on the remote server being accessed, although it is still valid on the original, authenticating server.

Authentication timeout processing on each server occurs after the time specified on that server. This time is specified by the SET WEBAUTHTIMEOUT command on each server. This timeout value must be consistent across all servers because an authentication date and time stamp, in Grenwich Mean Time (GMT) format, is included in the authentication token. Inconsistent settings can result in administrators having to reauthenticate more frequently than is desired.

Administrator authority is not passed between servers. An administrator userid might have SYSTEM authority on one server but only OPERATOR authority on another. This scenario does not prevent the userid from being used for single administrative login purposes. When a userid is used to access a remote server, it inherits the authority assigned to it on the server being accessed, not the authority it had on the server where it originally authenticated. If administrative access is required for multiple servers with consistent levels of authority, these authorities must be set on all servers along with the userid and password.

Administrator definitions for single administrative login can be created and maintained on multiple servers in an enterprise by exploiting the ADSM Enterprise Configuration functions (6.3, “Enterprise Configuration” on page 194). This involves defining the administrator userid, password, and authority on a configuration manager, associating that administrator userid with a configuration profile, and then subscribing the remote servers to that profile. Thus the administrator userid used for single administrative login is propagated to all remote managed servers. Any future updates required for the administrator userid could then be made centrally, and the changes would be automatically propagated to the remote servers.

7.2.1.2 Define Server URL

To hyperlink to a remote server, a server definition for the remote server is required (6.1, “Server-to-Server Communications” on page 178). The server definition created using the DEFINE SERVER command contains an optional URL parameter that must be specified to create the hyperlink.

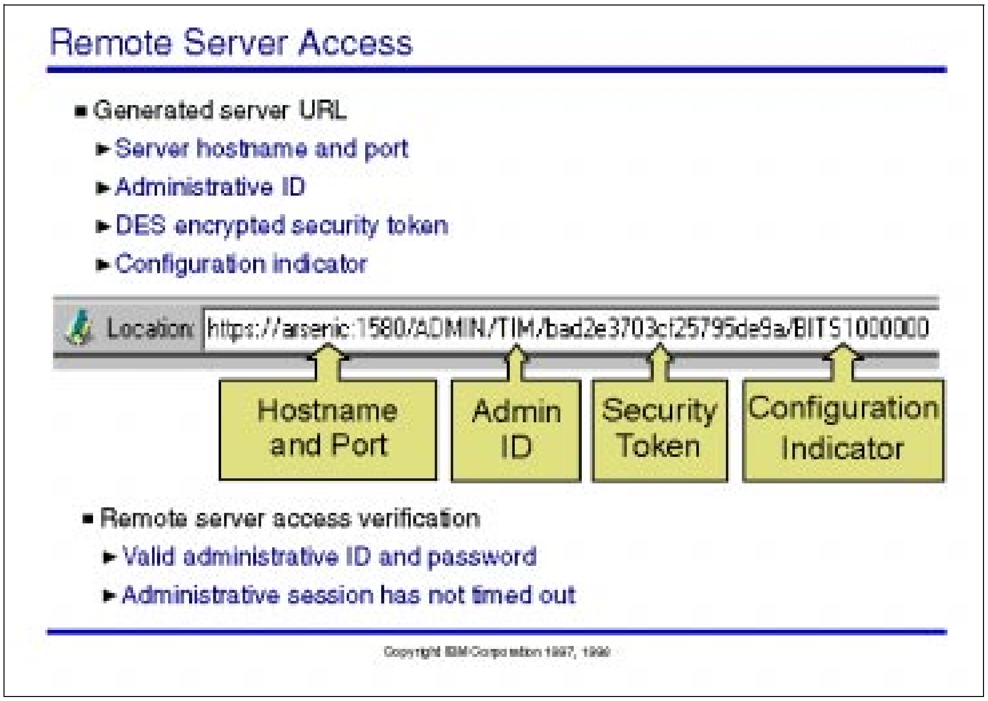
The URL parameter can be specified when issuing the DEFINE SERVER command to define the server. If the server has previously been defined, the UPDATE SERVER command, with the URL=*url_address* parameter, can be used to add the URL address to an existing server definition:

```
update server adsm_arsenic url=https://arsenic:1580
```

The URL is prefixed with either HTTP for the standard browser protocol or HTTPS for the SSL browser protocol. This protocol prefix can be omitted when defining the URL. However, if it is omitted, when the administrator follows the hyperlink, the same protocol that was used to start the Web administrative interface will be used for the hyperlink connection. This is particularly important if SSL is being used. Moving from one server where SSL is used to another where SSL is not used will fail if the URL allows the protocol to default to HTTPS.

7.2.1.3 Server Hyperlink Icon

Defining a URL for a server creates a hyperlink icon for that server in the Network view of the Web Enterprise Console. Clicking on this icon constructs a URL consisting of the server address, the administrator userid being used, a DES encrypted security token, and a server filename. This constructed URL is used to access the remote server.



Remote Server Access

- Generated server URL
  - Server hostname and port
  - Administrative ID
  - DES encrypted security token
  - Configuration indicator

Location: https://arsenic:1580/ADMIN/TIM/bad2e3703cf25795de9a/BITS1000000
Hostname and Port
Admin ID
Security Token
Configuration Indicator

- Remote server access verification
  - Valid administrative ID and password
  - Administrative session has not timed out

Copyright IBM Corporation 1997, 1998

7.2.2 Remote Server Access

Clicking on a server hyperlink icon in the network view of the Enterprise Console accesses that ADSM server. The generated URL contains all information required by the remote server to verify that the administrator userid is authorized to access the server.

7.2.2.1 Generated Server URL

The URL generated by clicking on a server hyperlink icon contains the following information:

- The HTTP, or HTTPS for SSL protocol, address and port number of the server to be accessed.
- The administrator userid being used. The /ADMIN prefix indicates that an administrator userid and security token are the next elements in the URL. In the example in the graphic an administrator userid of TIM is used.
- A DES encrypted security token that is used by the remote server to verify that the administrator userid is authorized to connect to the server. The password used to authenticate the administrator userid on the authenticating server is also used as part of the encryption key to generate the security token.
- A configuration indicator used by the remote server to format the Web administrative interface displayed.

7.2.2.2 Remote Server Access Verification

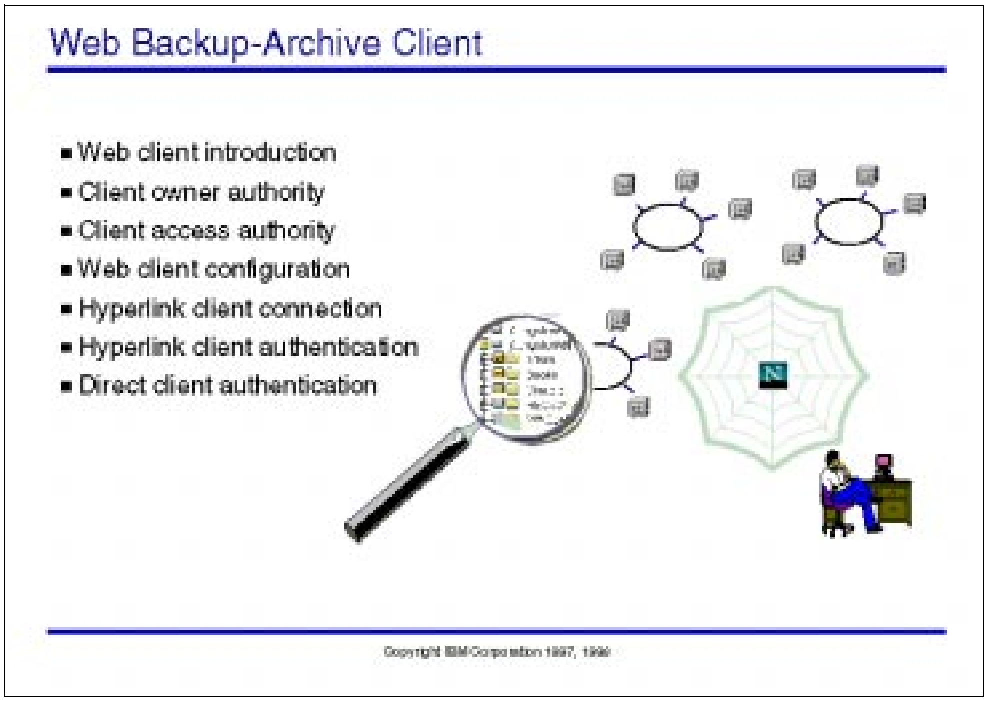
When the remote server is accessed with a generated URL, it first validates that the administrator userid and password are the same on both servers. The administrator userid is passed as part of the URL. The server being accessed then uses the locally defined password for that userid to decrypt the security token received as part of the URL. Successful decryption of the security token validates that the passwords for the administrator userid being used are the same on both servers.

The decrypted security token contains a date and time stamp of when the administrator userid was authenticated on the original authenticating server. This information is used to verify that the administrator login has not expired, based on the local WEBAUTHTIMEOUT server setting. The date and time stamps used are based on GMT to avoid complications that could arise from accessing servers in multiple time zones.

If the remote server cannot decrypt the security token passed to it, the login fails and a new administrative login prompt is displayed. If the remote server successfully decrypts the security token, using the locally defined password, but the login has expired, the administrator is again prompted to reauthenticate.

If the administrative login has not timed out, the hyperlink to the server is successful and the primary panel of the ADSM Enterprise Console is displayed for the remote server, without the administrator having to reauthenticate the userid. The administrator authority for the remote server will be the authority assigned to that administrator userid on that server, which may or may not be the same as on the server where the administrator originally authenticated.

7.3 Web Backup-Archive Client

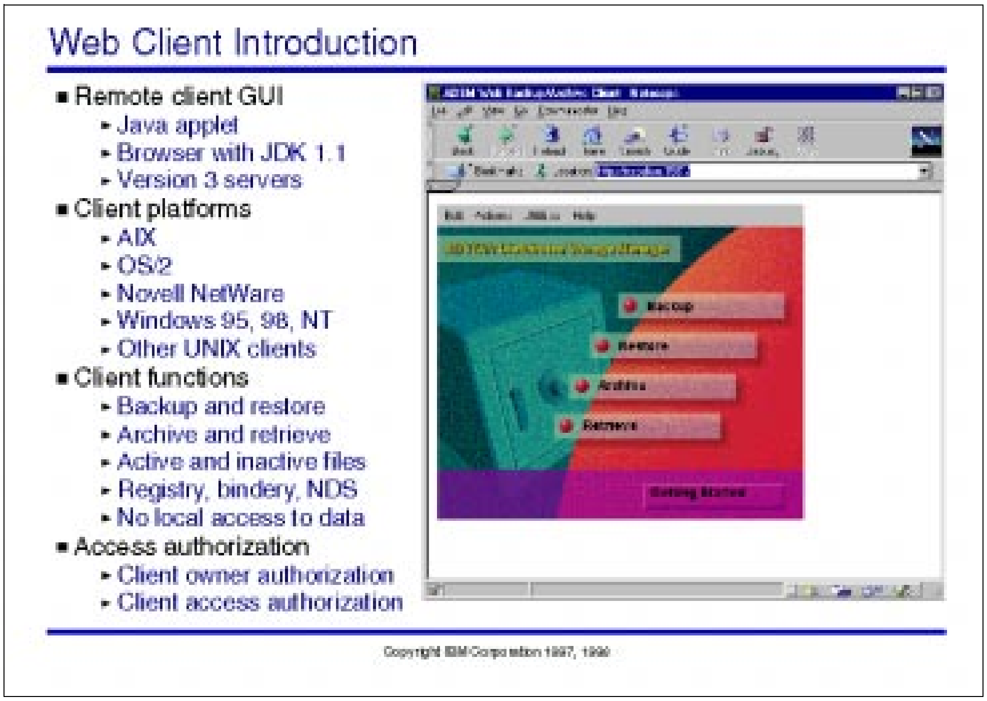


Web Backup-Archive Client

- Web client introduction
- Client owner authority
- Client access authority
- Web client configuration
- Hyperlink client connection
- Hyperlink client authentication
- Direct client authentication

This section covers the Web backup-archive client interface introduced with ADSM Version 3.1.2. The following topics are covered:

- Web client introduction
- Client owner authority
- Client access authority
- Web client configuration
- Hyperlink client connection
- Hyperlink client authentication
- Direct client authentication



Web Client Introduction

- Remote client GUI
  - Java applet
  - Browser with JDK 1.1
  - Version 3 servers
- Client platforms
  - AIX
  - OS/2
  - Novell NetWare
  - Windows 95, 98, NT
  - Other UNIX clients
- Client functions
  - Backup and restore
  - Archive and retrieve
  - Active and inactive files
  - Registry, bindery, NDS
  - No local access to data
- Access authorization
  - Client owner authorization
  - Client access authorization

Copyright IBM Corporation 1997, 1999

7.3.1 Web Client Introduction

The ADSM Version 3 Web backup-archive client is a Java client interface that enables authorized users to perform remote operations on a backup-archive client system.

7.3.1.1 Remote Client GUI

The Web client is a Java applet that provides a remote client GUI for ADSM Version 3 backup-archive clients. Authorized users can access the Web client remotely using a Web browser. The graphic is the initial client hub window displayed in the Web browser when the Web client is accessed.

The Web client can be used from any workstation running a Web browser with support for Java 1.1.5 or higher. Netscape 4.06 has the required Java support. Netscape 4.03, 4.04, and 4.05 requires a JDK 1.1 upgrade for this support. This upgrade can be obtained from the Netscape homepage. Microsoft's Internet Explorer 4.01 has the required Java support. If the browser does not have the correct level of Java support, the client hub window is displayed as a grey box.

The Web client is supported only on ADSM Version 3 servers.

7.3.1.2 Client Platforms

The Web client is supplied as PTF6 to the existing ADSM Version 3 clients. The following client platforms are supported:

- AIX 4.1.4, 4.1.5, 4.2.1, and 4.3.1
- OS/2 Warp 3.0 and 4.0
- NetWare 3.12 and 4.11
- Windows 95, Windows 98, and Windows NT 4.0 (Intel only)
- Digital UNIX 4

- HP-UX 11.0
- SGI IRIX 6.5
- Solaris 2.6

The Web client is a subcomponent of the backup-archive client package. The native backup-archive clients can be installed with the Web client interface. The Web client interface is not a separately installed package as was the case with the previous Webshell client, which is replaced by this new Web client.

7.3.1.3 Client Functions

The Web client supports most of the Version 3 backup-archive client functions. It can be used to perform backup, restore, archive, and retrieve operations. For restore operations, both active and inactive files can be restored, and the point-in-time restore function is supported. The functions not currently supported include cross client restores, filespace deletion, archive file deletion, and the file find and search functions.

The Windows NT Web client can back up and restore the NT registry. The NetWare Web client can backup and restore the NetWare 3.x bindery and NetWare 4.x Directory Services (NDS) and provides for the first time a NetWare client GUI.

The Web client is another client interface. It can be used only to perform backup-archive client operations. If used from a Web browser on a remote workstation, no local access to backup or archive data is provided on that remote workstation. Data cannot be restored or retrieved locally; it can only be restored to or retrieved to the client workstation that owns the data. A Web browser can also be used locally on the client workstation to invoke the Web client as an alternative interface to the backup-archive client command line interface or GUI.

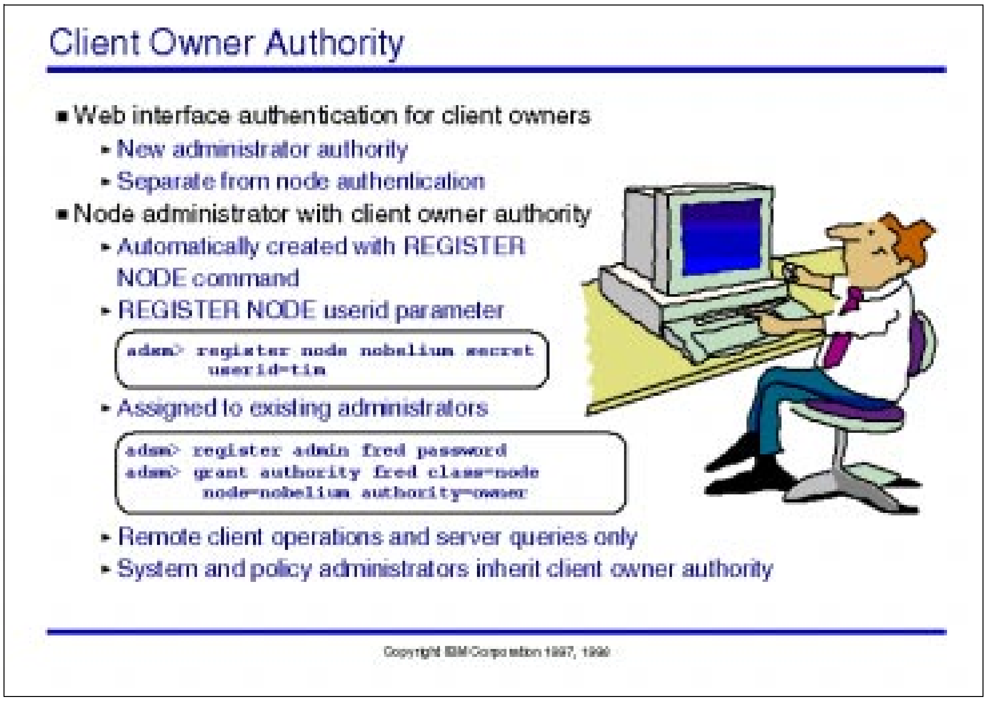
7.3.1.4 Access Authorization

The Web client is a new interface in addition to the client’s existing GUI and command line interface. Use of the Web client interface is authenticated whenever backup, restore, archive, or retrieve Web client functions are performed. Authentication of the Web client interface is separate, and independent, from authentication between the client node and the server.

An administrator userid is required to use the Web client. This administrator userid, and associated password, is used to authenticate that the user has sufficient authority to perform remote client functions. Two new administrative authorities are provided to enable this authentication:

- Client owner
- Client access

These new authorities are available with the Version 3.1.2.1 PTF level of the ADSM server and can be used to enable usage of the Web client interface for backup-archive client owners and helpdesk personnel.



Client Owner Authority

- Web interface authentication for client owners
  - New administrator authority
  - Separate from node authentication
- Node administrator with client owner authority
  - Automatically created with REGISTER NODE command
  - REGISTER NODE userid parameter

```
adsm> register node nobelium secret
        userid=tim
```

  - Assigned to existing administrators

```
adsm> register admin fred password
adsm> grant authority fred class=node
        node=nobelium authority=owner
```

  - Remote client operations and server queries only
  - System and policy administrators inherit client owner authority

Copyright IBM Corporation 1997, 1999

7.3.2 Client Owner Authority

An administrator userid can be defined with client owner authority to provide a userid, independent of the client node name and password, for use by the owner of the client to access the Web client interface.

7.3.2.1 Web Interface Authentication for Client Owners

Use of the Web client is authenticated. Clicking on any client function button (backup, restore, archive, or retrieve) requires that an administrator userid and password be used to authenticate use of the interface. Client owner is a new level of authorization that can be applied to an administrator userid for this purpose.

Every time a backup-archive client connects to the server, it authenticates the session, using the current password for that node. For the Web client, this client password handling must be automated using a generated password created by specifying the PASSWORDACCESS GENERATE option in the client options file. Authentication of the Web client interface is separate from this client authentication with the server. If the owner of the client wants to use the Web client interface he or she must also have an administrator userid and password. The new client owner authority level is provided to enable creation of such administrator userids.

7.3.2.2 Node Administrator with Client Owner Authority

With Version 3.1.2.1 of the ADSM server, whenever a client node is registered with the REGISTER NODE command, an administrator userid, with the same name and password as the node name, is automatically defined:

```
adsm> register node nobelium secret
ANR2060I Node NOBELIUM registered in policy domain STANDARD.
ANR2099I Administrative userid NOBELIUM defined for OWNER access to node NOBELIUM
```

This command registers a node named *nobelium* with a password of *secret,* and an administrator userid, with client owner authority, is also created with the same name and password. The REGISTER NODE command has a new userid parameter for defining an administrator userid with client owner authority. The parameter is optional. If a userid is not specified, as in the example above, an administrator userid is defined with the same name as the node name. The userid parameter can be specified to create an administrator userid of a different name from that of the node being registered:

```
adsm> register node nobelium secret userid=tim
ANR2060I Node NOBELIUM registered in policy domain STANDARD.
ANR2099I Administrative userid TIM defined for OWNER access to node NOBELIUM
```

In this example the *tim* administrator userid would have the same password as the *nobelium* client node. If the *tim* administrator userid were already defined, the REGISTER NODE command would just add client owner authority for the *nobelium* node to the existing privileges of the *tim* administrator userid. To disable automatic creation of administrator userids, a *userid=none* parameter must be specified.

When the REMOVE NODE command is used to remove a node definition, the administrator userid with client owner authority for the node being removed is also updated. If the administrator userid has only client owner authority for the node being removed, the administrator userid is also removed:

```
adsm> remove node nobelium

Do you wish to proceed?y
ANR2061I Node NOBELIUM removed from policy domain STANDARD.
ANR2129I Administrative userid NOBELIUM defined ONLY for authority over node
NOBELIUM has been removed.
```

If the administrator userid has other authorities such as system, policy, or client owner authority for other nodes in addition to the node being removed, only the client owner authority for the node is removed from that administrator userid. The administrator userid is not removed.

The client owner authority can be assigned to existing administrators through the GRANT AUTHORITY command. This command has a new class parameter of *node* and new *node* and *authority* options:

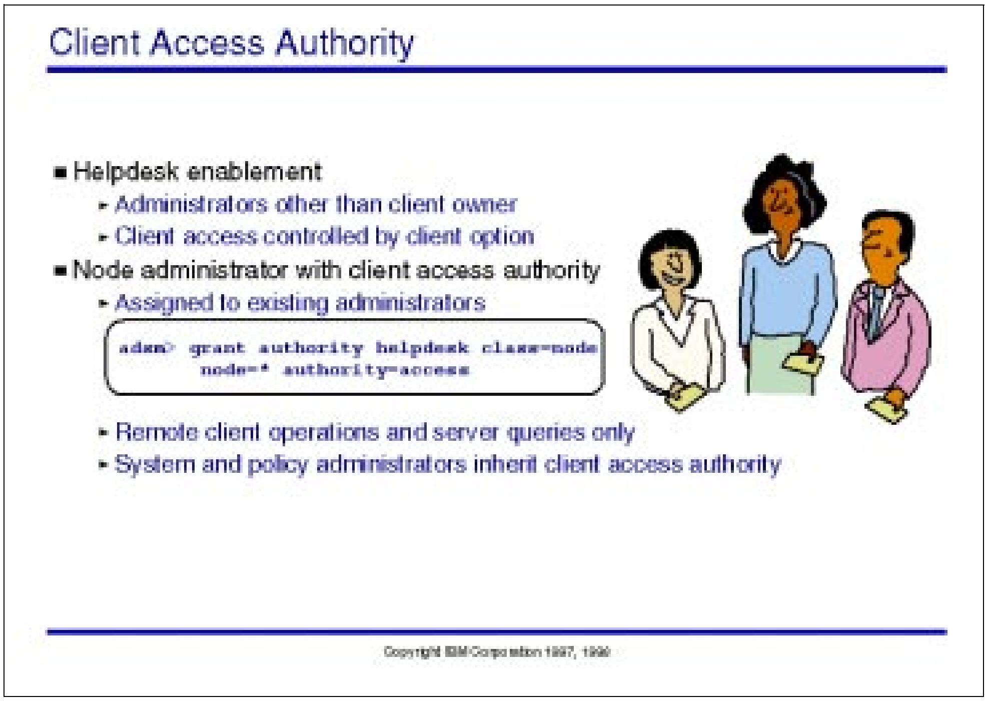
```
adsm> register admin fred password
adsm> grant authority fred class=node node=nobelium authority=owner
```

These two commands first create an administrator userid of *fred*, and then secondly, assign to it client owner authority for the *nobelium* node. The *class* parameter specifies that the administrator is a client node administrator. The *node* parameter determines which client node the administrator is authorized to access, and the *authority* parameter defines the authority level, in this case, client owner. The node parameter can also be specified with a wildcard, node=*, to assign the authority to all registered nodes. The QUERY ADMIN command reflects these new authorities:

```
adsm> query admin

Administrator       Days Since     Days Since     Locked?      Privilege Classes
Name               Last Access   Password Set
--------------    ------------   ------------   ----------   -----------------
ADMIN                        4              5         No      System
FRED                        <1             <1         No      Client Owner
NOBELIUM                     5              5         No      Client Owner
TIM                          5              5         No      Client Owner
```

Client owner administrator userids can perform Web client operations. As these are administrator userids they can also be used to log in to the ADSM server. However, they are limited to performing server queries. They cannot perform updates to the server.

Existing administrators with system or policy authority automatically inherit client owner authority for all registered nodes. In the case of restricted policy administrators, the userid only has client owner authority for those nodes registered in the policy domains in which they have policy authority.



Client Access Authority

- Helpdesk enablement
  - Administrators other than client owner
  - Client access controlled by client option
- Node administrator with client access authority
  - Assigned to existing administrators

```
adsm> grant authority helpdesk class=node
        node=* authority=access
```

  - Remote client operations and server queries only
  - System and policy administrators inherit client access authority

Copyright IBM Corporation 1997, 1998

7.3.3 Client Access Authority

An administrator userid can be defined with client access authority to provide administrators or helpdesk personnel, other than the client owner, with access to the Web clients.

7.3.3.1 Helpdesk Enablement

The client access authority is provided to enable access to the Web client for administrators other than the client owner, such as helpdesk personnel. An administrator userid with client access authority can perform the same Web client functions as an administrator userid with client owner authority. Access to Web clients by administrators with client access authority is controlled by a client option defined in the client options file.

7.3.3.2 Node Administrator with Client Access Authority

The client access authority is assigned to existing administrators in the same manner as the client access authority by using the GRANT AUTHORITY command with *class*, *node*, and *authority* parameters:

```
adsm> register admin helpdesk password
adsm> grant authority helpdesk class=node node=* authority=access
```

This example registers an administrator userid named helpdesk and assigns to it client access authority for all registered nodes.

Client access and client owner authorities are mutually exclusive for nodes associated with administrator userids. For a given node name, an administrator userid can have client owner *or* client access authority; it cannot have both. For example, if an administrator userid is automatically created through the REGISTER NODE command, it will automatically have client owner authority for that node. If the GRANT AUTHORITY command is then used to assign client access authority for the same node name to the administrator userid, the original client owner authority is revoked and replaced with client access authority.

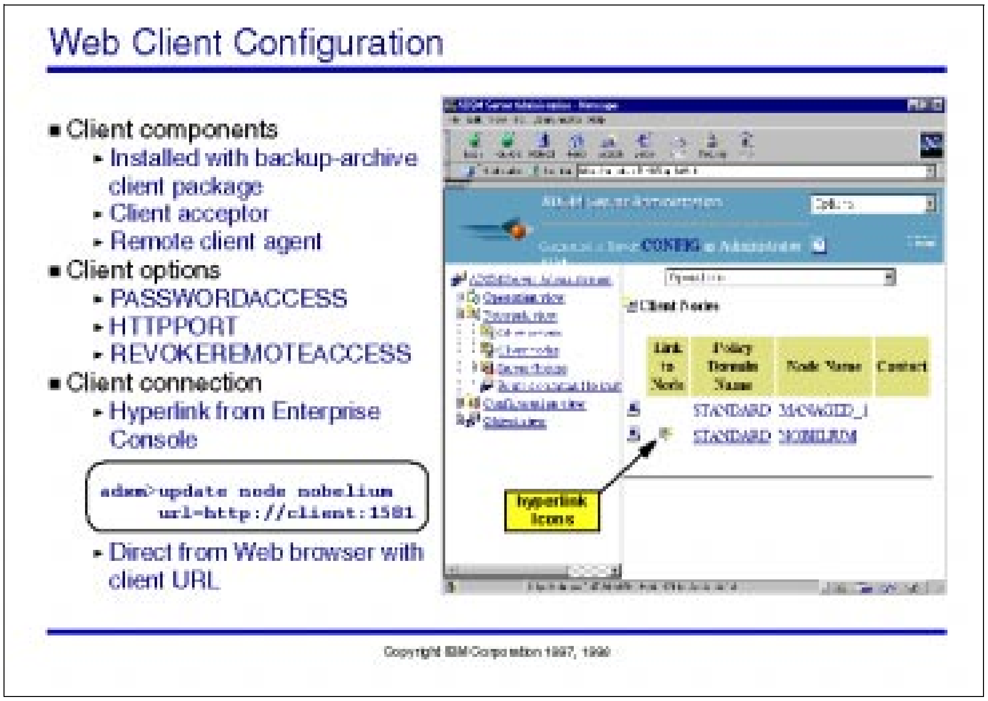
The QUERY ADMIN command displays the client access authority:

```
adsm> query admin

Administrator       Days Since      Days Since     Locked?      Privilege Classes
Name               Last Access    Password Set
--------------    ------------    ------------    ----------    -----------------
ADMIN                        4               5        No        System
FRED                        <1              <1        No        Client Owner
HELPDESK                    <1              <1        No        Client Access
NOBELIUM                     5               5        No        Client Owner
TIM                          5               5        No        Client Owner
```

Client access administrator userids can perform the same Web client operations as client owner administrator userids. The only difference between the authorities is that a client option can be used to restrict Web client access from administrators with only client access authority.

As with administrators with client owner authority, these administrator userids can also be used to log in to the ADSM server. However, they are limited to performing server queries. They cannot perform updates to the server.

Existing administrators with system or policy authority automatically inherit client access authority for all registered nodes. In the case of restricted policy administrators, the userid only has client access authority for those nodes registered in the policy domains in which they have policy authority.



Web Client Configuration

- Client components
  - Installed with backup-archive client package
  - Client acceptor
  - Remote client agent
- Client options
  - PASSWORDACCESS
  - HTTPPORT
  - REVOKEREMOTEACCESS
- Client connection
  - Hyperlink from Enterprise Console

```
adsm>update node nobelium
       url=http://client:1581
```

  - Direct from Web browser with client URL

7.3.4 Web Client Configuration

The Web client consists of new client components that must be installed and configured before the client can be accessed with a Web browser.

7.3.4.1 Client Components

The Web client is installed with the backup-archive client package. It is not a separate package as was the Webshell client that it replaces. The Web client consists of two new processes on the client workstation: the client acceptor and remote client agent.

The client acceptor is an HTTP daemon that serves the Web client Java applet to the Web browser. The name of the executable is DSMCAD. On AIX and other UNIX clients, it should be run as a daemon. For Windows NT it is installed and run as a service. For Windows 95 and 98 it must be executed in a DOS window. For NetWare it is an NLM that should be loaded as part of the NetWare startup.

The remote client agent performs the client functions initiated with the Web client interface. The name of the executable is DSMAGENT. The agent does not have to be running all the time. The acceptor daemon starts the agent when client functions are initiated through the Web client interface.

7.3.4.2 Client Options

Three client options are important for the Web client. The PASSWORDACCESS option must be set to GENERATE and the password must be generated by running a backup-archive client session. The remote client agent establishes connection to the ADSM in the same manner as the GUI and command line clients. It requires the generated client password to authenticate a client session when the Web client is used.

The acceptor daemon listens on a TCP/IP port for incoming connections from an administrator’s Web browser. By default it listens on port 1581. This default port can be overridden with the new HTTPPORT client option.

The new REVOKEREMOTEACCESS option has two possible values: NONE, which is the default, and ACCESS. If the option is set to NONE, any administrator userid with client access or client owner authority can perform client operations. If the option is set to ACCESS, administrator userids with only client access authority are prevented from performing remote client operations. A pop-up message displayed in the Web browser indicates that the administrator userid being used has insufficient authority. This option does not prevent administrators with client owner or higher authorities from performing client functions.

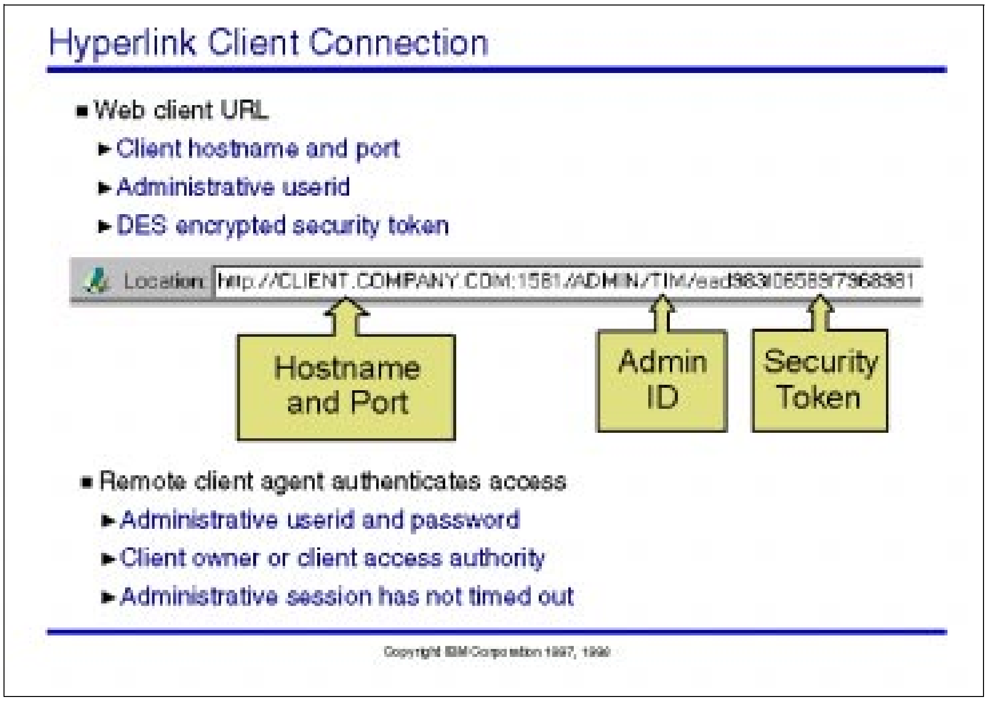
7.3.4.3 Client Connection

There are two methods of accessing the Web client from a Web browser. The first method involves creating a hyperlink from the Web administrative Enterprise Console by defining a URL for the Web client:

```
adsm> update node nobelium url=http://client:1581
```

Defining a client URL creates a client hyperlink icon. Clicking on this hyperlink icon connects to the Web client and exploits the single administrative login functions.

The second method is to connect directly from a Web browser by entering the URL and port number of the client in the location field of the browser. When using this method, the user is prompted for a userid and password when any client backup, restore, archive, or retrieve functions are initiated. For Version 3.1.2 or later servers, an administrative userid and password are used. For Version 3 servers prior to 3.1.2, the nodename and node password must be entered.



Hyperlink Client Connection

- Web client URL
  - Client hostname and port
  - Administrative userid
  - DES encrypted security token

Location:
http://CLIENT.COMPANY.COM:1581/ADMIN/TIM/ead983I06589I7968981
Hostname and Port
Admin ID
Security Token

- Remote client agent authenticates access
  - Administrative userid and password
  - Client owner or client access authority
  - Administrative session has not timed out

7.3.5 Hyperlink Client Connection

Connecting to a Web client from an Enterprise Console hyperlink icon exploits the single administrative login functions in the same way as when accessing another ADSM server (7.2, “Server Single Administrative Login” on page 221).

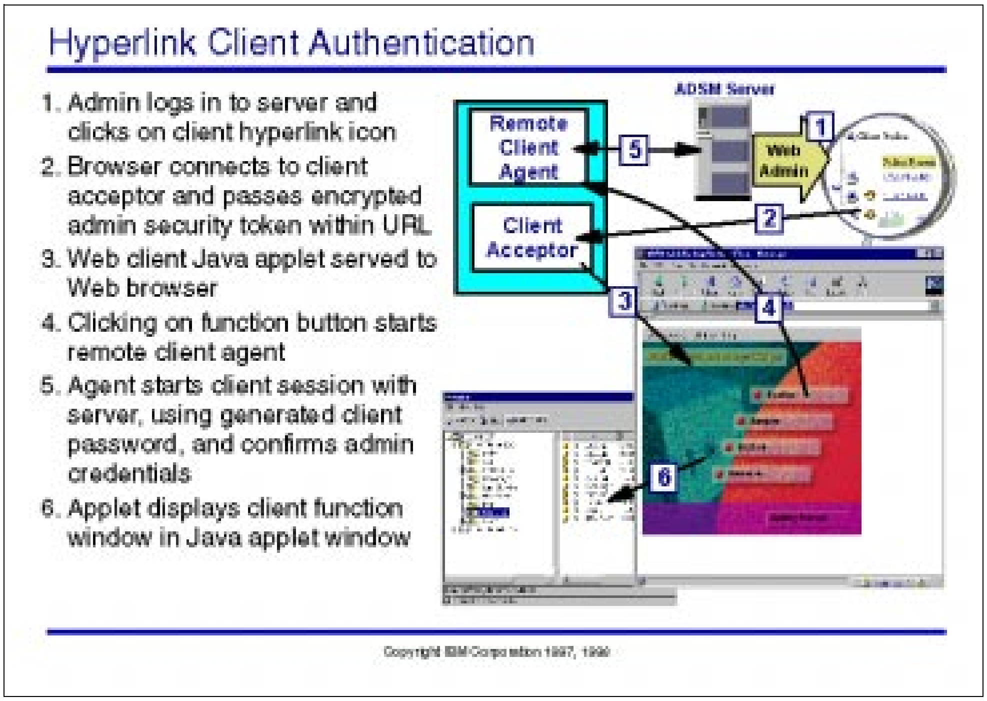
7.3.5.1 Web Client URL

Clicking on a client hyperlink icon in the Network view of the Enterprise Console generates a URL that contains the client host name and port, the administrative userid being used, and a DES encrypted security token. The port is the port number on which the client acceptor daemon is listening for incoming Web browser connections.

7.3.5.2 Remote Client Agent Authenticates Access

When a Web client function such as backup, restore, archive, or retrieve is selected, the remote client agent is invoked on the client workstation. The remote client agent connects to the ADSM server and verifies that the administrative userid being used is authorized to perform Web client functions. It decrypts the DES security token passed as part of the URL, using the administrator password on the ADSM server. If the decryption is successful, it determines that the administrative userid has client access, client owner, system, or policy privileges. Finally it determines that the administrative session has not timed out. If these three checks are successful, the client function window is displayed.

The next two graphics illustrate the connection and authentication sequence for a hyperlink connection and direct URL connection to the Web client.



Hyperlink Client Authentication

1. Admin logs in to server and clicks on client hyperlink icon
2. Browser connects to client acceptor and passes encrypted admin security token within URL
3. Web client Java applet served to Web browser
4. Clicking on function button starts remote client agent
5. Agent starts client session with server, using generated client password, and confirms admin credentials
6. Applet displays client function window in Java applet window

Copyright IBM Corporation 1997, 1999

7.3.6 Hyperlink Client Authentication

When the Web client is accessed from the Enterprise Console, the authentication to use the interface is transparent. The administrator userid and security token are passed to the client. The client uses these to authenticate the user with the ADSM server. The following sequence takes place when a Web client function is initiated from a hyperlink icon on the Enterprise Console:

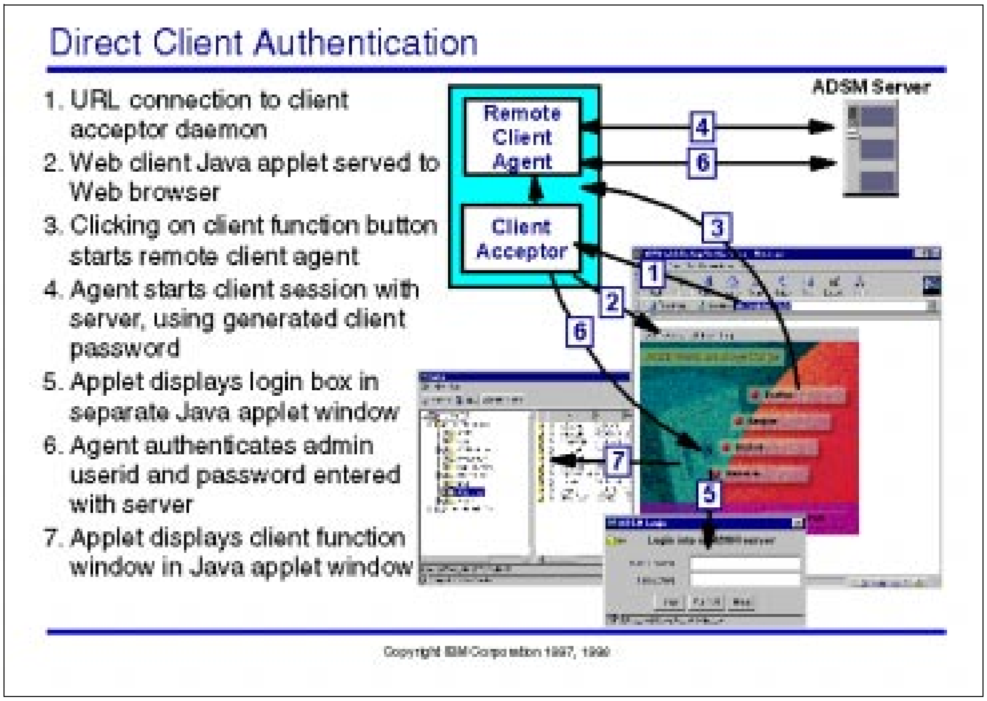
1. The administrator logs in to the ADSM server, using the Web administrative interface. Selecting the *Client nodes* option from the Network view of the Enterprise Console displays a list of the defined clients. Clicking on a client hyperlink icon generates a URL and connects to the client.
2. The Web browser connects to the client acceptor daemon, using the generated URL that contains the administrative userid being used and a DES encrypted security token for that administrator session.
3. The client acceptor daemon serves the Web client Java applet back to the Web browser. The client hub window is displayed in the main window of the Web browser.
4. The client hub window is displayed without any client or administrator authentication taking place. When a client function is selected from the client hub, the Java applet contacts the client acceptor daemon and the client acceptor ensures that the remote client agent is running on the client workstation. The remote agent performs authentication for the client and the administrator userid invoking the Web client function.
5. The remote client agent starts a client session with the ADSM server. This is a client session using the client nodename and generated password. The agent must be able to start this client session before performing any further functions. When the agent has successfully started a client session it uses this client session to pass the administrator userid and DES encrypted security

token to the ADSM server. The ADSM server verifies that the administrative userid and password are valid, that the administrator userid has client access or client owner authority, and that the administrative session has not timed out.

Client Access Authority

If the administrator userid being used has only client access authority and the client has the REVOKEREMOTEACCESS client option set to ACCESS, the administrator authentication will fail. A message displayed in a Java applet window indicates that the administrator userid has insufficient authority to perform the client operation.

6. If the administrator userid is successfully authenticated, the client function window is displayed in a separate Java applet window. The remote client function has now been authenticated and can be performed.



Direct Client Authentication

1. URL connection to client acceptor daemon
2. Web client Java applet served to Web browser
3. Clicking on client function button starts remote client agent
4. Agent starts client session with server, using generated client password
5. Applet displays login box in separate Java applet window
6. Agent authenticates admin userid and password entered with server
7. Applet displays client function window in Java applet window

Copyright IBM Corporation 1997, 1999

7.3.7 Direct Web Client Authentication

When the Web client is accessed directly from a Web browser, an administrator userid must be entered to authenticate the Web client interface before any client functions can be performed. The client authenticates with the server that the user has the appropriate authority. The following sequence takes place when a Web client function is initiated:

1. The Web browser connects to the client acceptor daemon, using the URL of the client workstation. This URL contains the hostname and the port on which the client acceptor daemon is listening for incoming Web browser connections.
2. The client acceptor daemon serves the Web client Java applet back to the Web browser. The client hub window is displayed in the main window of the Web browser.
3. The client hub window is displayed without any client or administrator authentication taking place. When a client function such as backup is selected from the client hub, the Java applet starts the remote client agent on the client workstation. The remote agent performs client authentication and administrator authentication to perform the Web client function.
4. The remote client agent starts a client session with the ADSM server. This is a client session using the client nodename and generated password. The agent must be able to start a client session before performing any further functions.
5. When the agent has successfully started a client session, it displays a login prompt in a separate Java applet window on the Web browser. An administrator userid and password must be entered to authenticate use of the Web client function.
6. The Java applet and the remote client log in to the ADSM server, using the userid and password entered. The ADSM server verifies that the

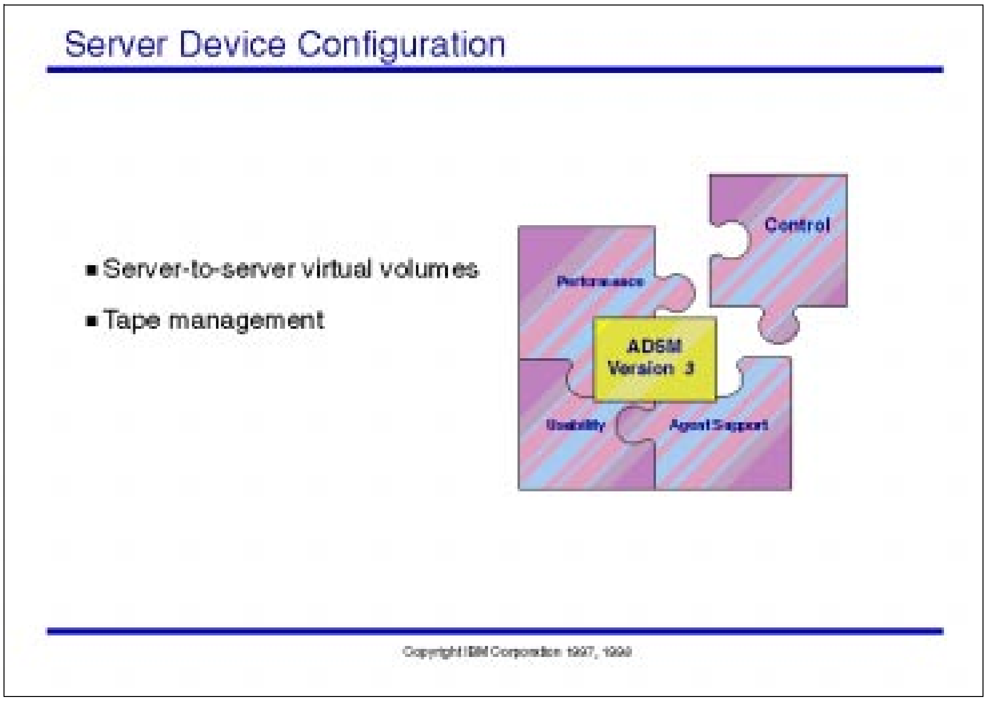
administrative userid and password are valid, that the administrator userid has client access or client owner authority, and that the administrative session has not timed out.

Version 3.1.1 Servers

With an ADSM Version 3.1.2 or later level of server an administrator userid and password must be entered in the login Java applet window. With Version 3.1.1 (non-Enterprise Management) servers, the client *nodename* and *password* must be entered.

7. If the administrator userid is successfully authenticated, the client function window is displayed in a separate Java applet window. The remote client function has now been authenticated and can be performed.

Chapter 8. Server Device Configuration



Server Device Configuration

- Server-to-server virtual volumes
- Tape management

Control
Performance
ADSM
Version 3
Usability
Agent Support

Copyright IBM Corporation 1997, 1998

This chapter covers new functions introduced in ADSM Version 3 for server device configuration. It covers the following topics:

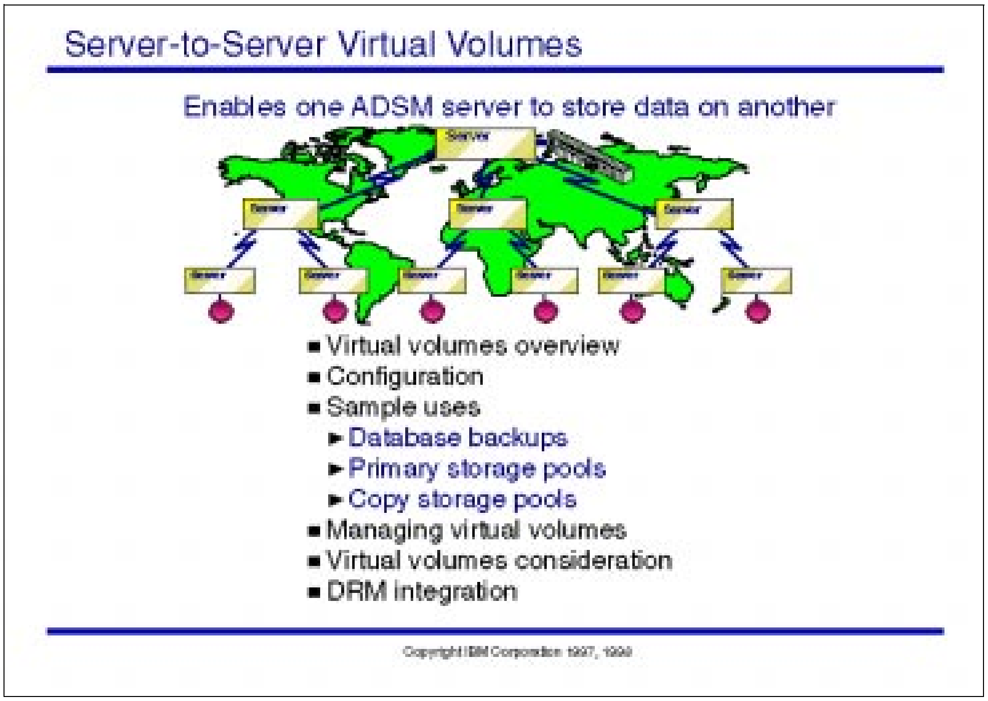
- Server-to-server virtual volumes

  This section shows how to configure server-to-server communications between two servers to use server-to-server virtual volumes.

- Tape management

  This section covers overflow storage pools, storage pool migration by age, mount limit management, single drive reclamation, and tape labeling.

8.1 Server-to-Server Virtual Volumes



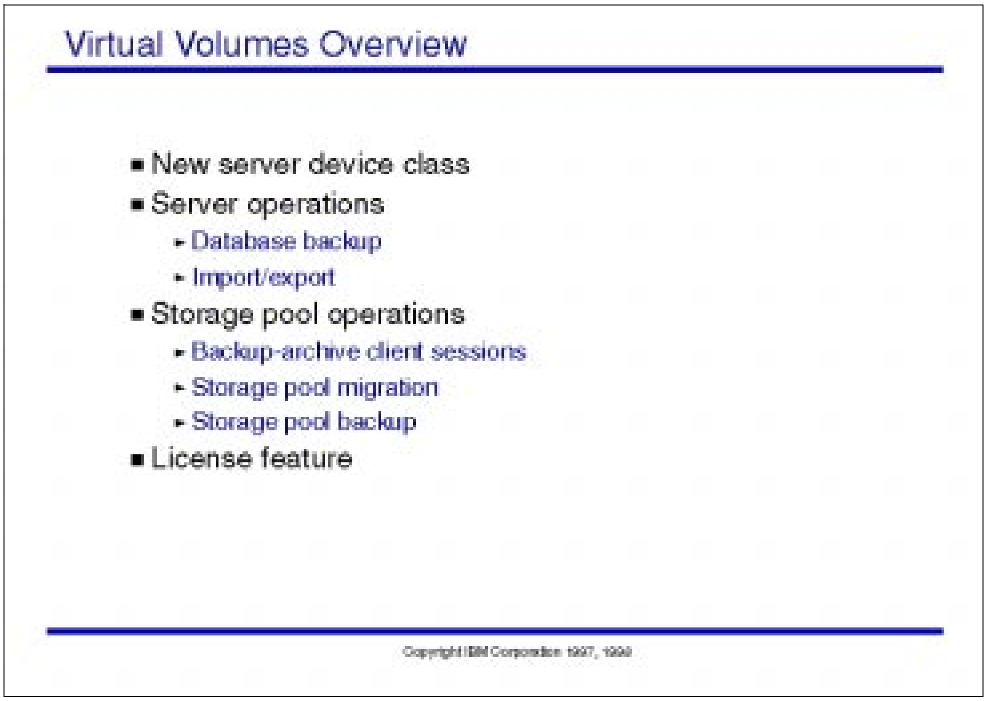
Server-to-Server Virtual Volumes

Enables one ADSM server to store data on another

- Virtual volumes overview
- Configuration
- Sample uses
  - Database backups
  - Primary storage pools
  - Copy storage pools
- Managing virtual volumes
- Virtual volumes consideration
- DRM integration

The objective of server-to-server virtual volumes is to enable one ADSM server to store data on another. This section provides:

- An overview of server-to-server virtual volumes
- A three step example of enabling server-to-server virtual volumes
- Examples of how server-to-server virtual volumes can be used for server database backups and primary and copy storage pools
- An overview of how data is stored in virtual volumes when they are used between two servers
- Considerations for managing server-to-server virtual volumes
- Disaster Recovery Manager (DRM) integration with server-to-server virtual volumes.



Virtual Volumes Overview

- New server device class
- Server operations
  - Database backup
  - Import/export
- Storage pool operations
  - Backup-archive client sessions
  - Storage pool migration
  - Storage pool backup
- License feature

8.1.1 Virtual Volumes Overview

Server-to-server virtual volumes provide a flexible infrastructure for connecting multiple ADSM Version 3 servers.

8.1.1.1 New Server Device Class

With ADSM Version 2, a device class uses a directory for file media, or a library for tape or optical media. With ADSM Version 3, a device class can point to a remote, network attached ADSM server.

8.1.1.2 Server Operations

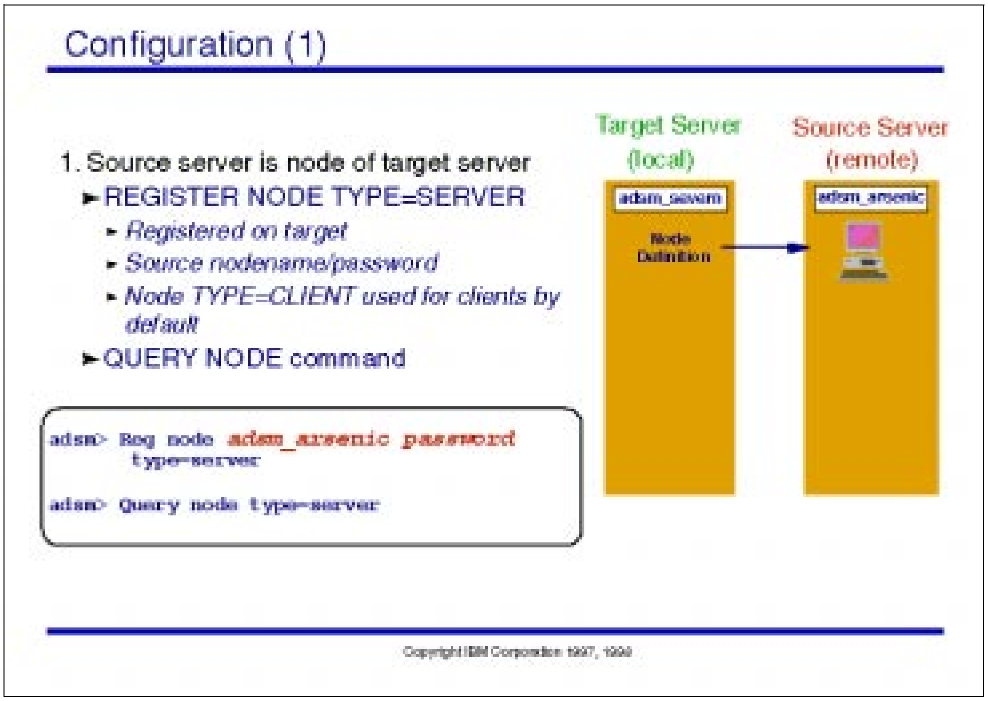
With a server device class, server-to-server virtual volumes can be used for any operation that uses a device class, such as backup of the database or server import and export operations.

8.1.1.3 Storage Pool Operations

Because the remote server is accessed through use of a device class, a storage pool can be defined on the local server by using this device class. Storage pool operations such as client sessions (backup, archive, HSM), storage pool migration, and storage pool backup can use such a storage pool, which in turn uses the remote server.

8.1.1.4 License Feature

To use server-to-server virtual volumes, the *Server-to-Server Virtual Volumes* license must be registered on both servers by using the REGISTER LICENSE command.



Configuration (1)

1. Source server is node of target server
   - REGISTER NODE TYPE=SERVER
     - *Registered on target*
     - *Source nodename/password*
     - *Node TYPE=CLIENT used for clients by default*
   - QUERY NODE command

```
adsm> Reg node adsm_arsenic password
        type=server

adsm> Query node type=server
```

8.1.2 Configuration (1)

This and the next two graphics show the three steps required to enable server-to-server virtual volumes. This example illustrates the definitions required on both servers to enable server-to-server virtual volumes. The following terms are used to differentiate between the two servers shown in this sample configuration:

- *Source server* - The remote server on which the virtual volumes are created. The source server is typically the server to which the client connects, the first server in the server hierarchy.
- *Target server* - The local server where the virtual volumes are physically stored. Clients do not typically connect to this server. If the client chooses to connect to this server, it cannot restore or retrieve data that was backed up to the source server, even though the data may physically reside on the target server.

Server Terminology

The examples illustrated here are based on a hierarchical model of server definitions similar to that described in 6.1, “Server-to-Server Communications” on page 178. The terms *source* and *target* servers are used in this section to describe servers in the context of server-to-server virtual volumes and correspond to the terms *remote* and *local* servers, respectively.

8.1.2.1 REGISTER NODE TYPE=SERVER

The first step is to register the source server as a node on the target server. The source server is registered through the REGISTER NODE command with a new parameter, *type=server*. A node name and password are defined for the source server to use when communicating with the target server. Client workstations are still registered as nodes with the parameter *type=client*, which is the default when registering a node.

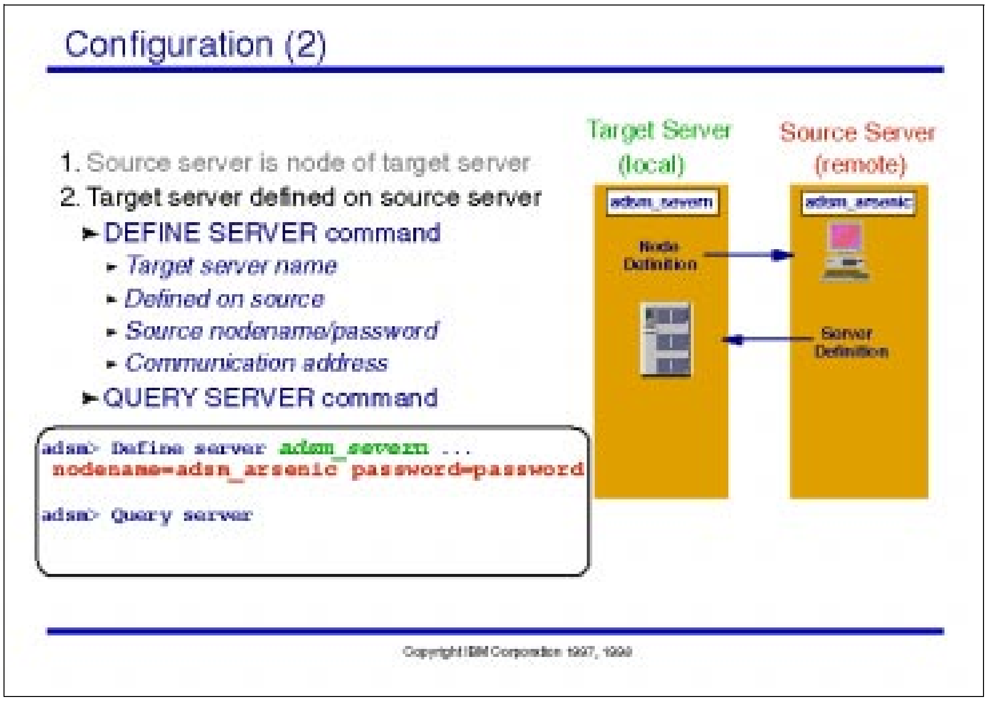
A node registered on an ADSM server with the parameter *type=server* cannot be registered on the same ADSM server with the parameter *type=client*.

8.1.2.2 QUERY NODE Command

Server nodes defined with the parameter *type=server* can be queried with the QUERY NODE command and the new *type=server* option on the target server:

```
adsm> query node type=server

Node Name         Platform     Policy Domain      Days Since    Days Since     Locked?
                                        Name            Last      Password
                                                      Access           Set
-------------     --------     --------------     ----------    ----------     -------
ADSM_ARSENIC      Windows      STANDARD                   <1            <1       No
                   NT
```

If the *type=server* option is not specified, only client nodes are displayed.



Configuration (2)

1. Source server is node of target server
2. Target server defined on source server
   - DEFINE SERVER command
     - *Target server name*
     - *Defined on source*
     - *Source nodename/password*
     - *Communication address*
   - QUERY SERVER command

```
adsm> Define server adsm_severn ...
 nodename=adsm_arsenic password=password

adsm> Query server
```

Copyright IBM Corporation 1997, 1998

8.1.3 Configuration (2)

The second step of the configuration is to define the target server as a server on the source server by using the DEFINE SERVER command. If server-to-server communications has already been defined, this step may not be required (see 6.1, “Server-to-Server Communications” on page 178).

8.1.3.1 DEFINE SERVER Command

On the source server, the target server must be defined as a server. When defining a server, you can specify the following additional parameters for the DEFINE SERVER command for virtual volumes:

DELGRACEPERIOD Specifies the number of days that an object remains on the target server after it has been marked for deletion. The minimum value is five days; there is no maximum. This is an optional parameter. The process used for deleting these archived objects is covered in “Managing Virtual Volumes” on page 256.

NODENAME Specifies the node name used by the source server to connect to the target server when a virtual volume is being used. This name must match the source server node name defined on the target server (“Configuration (1)” on page 244). If this parameter is not supplied, the name of the source server, as reported by a QUERY STATUS command, is used.

PASSWORD Specifies the node password used by the source server to connect to the target server. This parameter must be specified. It is the virtual volume password (which must match the password for the source server node) defined on the target server.

If a server definition has previously been created, these parameters can be added or modified by using the UPDATE SERVER command.

Nodename and Password

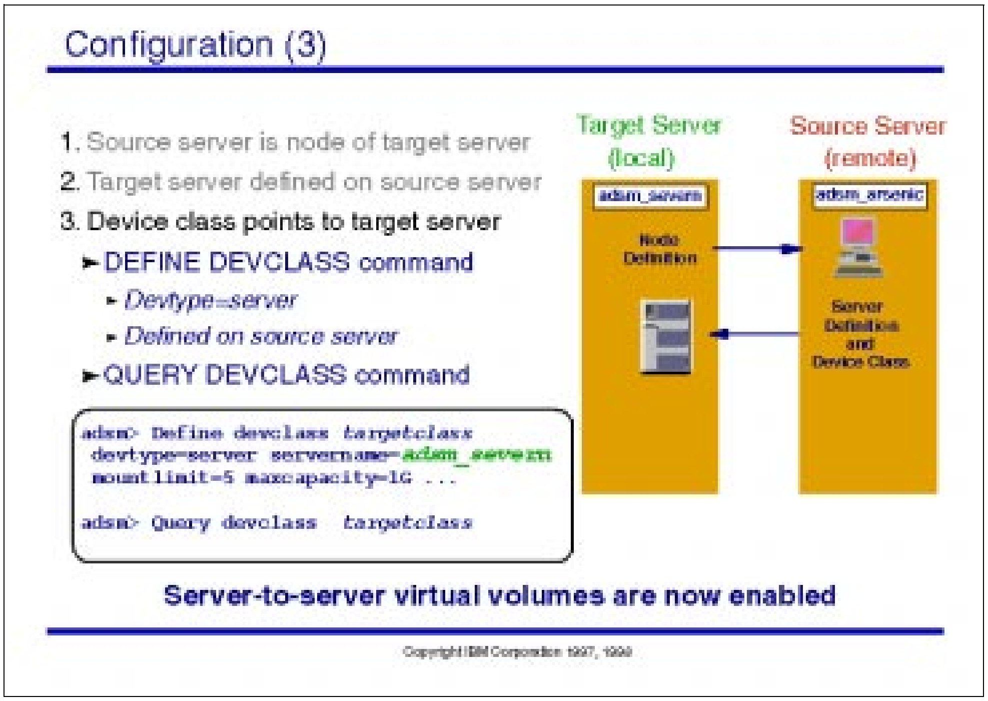
With ADSM Version 3.1.2, the DEFINE SERVER command has changed. Two separate passwords can now be defined. The SERVERPASSWORD is used to authenticate server-to-server sessions (6.1.5, “Server Definitions” on page 185). Authentication for server-to-server connections and virtual volumes is a separate process. The NODENAME and PASSWORD parameters specified here are used only for authenticating connections between the servers when virtual volumes are being used.

8.1.3.2 QUERY SERVER Command

The QUERY SERVER command can be used to query the server definitions on the source server:

```
adsm> query server

Server        Comm.  High-level    Low-level   Days Server   Virtual  Allow
Name          Method Address       Address    Since Password Volume   Replacement
                                               Last Set      Password
                                             Access          Set
-----------   ------ ------------- --------- ------ -------- -------- -----------
ADSM_SEVERN   TCPIP  9.1.2.3       1500           9 Yes      Yes      No
```

The command output shows the resulting server definition. The **Yes** in the Virtual Volume Password Set column indicates that the password has been successfully set. The server nodename can be displayed by using the format=detailed option.



Configuration (3)

1. Source server is node of target server
2. Target server defined on source server
3. Device class points to target server
   - DEFINE DEVCLASS command
     - *Devtype=server*
     - *Defined on source server*
   - QUERY DEVCLASS command

```
adsm> Define devclass targetclass
 devtype=server servername=adsm_severn
 mountlimit=5 maxcapacity=1G ...

adsm> Query devclass  targetclass
```

Server-to-server virtual volumes are now enabled

Copyright IBM Corporation 1997, 1998

8.1.4 Configuration (3)

The third step of the configuration is to define a device class on the source server, using the *devtype=server* parameter.

8.1.4.1 DEFINE DEVCLASS Command

When defining a device class on the source server, a *devtype=server* can be used. The following parameters can be used:

- SERVERNAME, to specify the name of the target server defined on the source server through the DEFINE SERVER command (“Configuration (2)” on page 246).
- MAXCAPACITY, to specify the maximum size of the objects that can be created on the target server. An object can be a storage pool virtual volume, database backup, or exported ADSM database data. The default is 500 MB. The amount of space specified by *maxcapacity* is reserved on the target server before the source server sends any data. Therefore, if this value is greater than the size of the target server storage pool, the data will not be stored on that storage pool. It will be directed to the *nextstoragepool* if one has been specified; otherwise the operation fails.
- MOUNTLIMIT, to specify the maximum number of simultaneous sessions that can be used between the source server and target server. If this parameter is not specified, it defaults to 1, which limits the number of server-to-server sessions to one.
- MOUNTRETENTION, to specify the amount of time to retain an idle virtual volume. If a client requests data that is held on a target server, from a source server, retaining the virtual volume mount can help to reduce the total time taken for communication to the client.

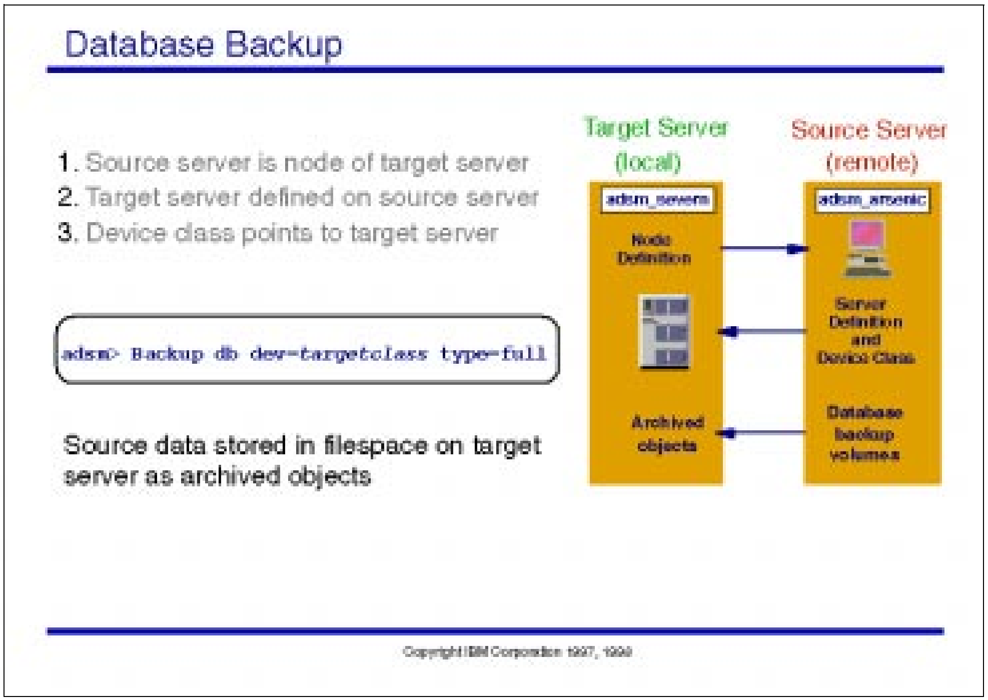
- PREFIX, to specify the beginning portion of the high-level archive file name on the remote server, to control the volume names that are generated
- RETRYPERIOD, to specify the retry period in minutes to restart a communication in the event of a communication failure
- RETRYINTERVAL, to specify how often the retries are done within the retry period

8.1.4.2 QUERY DEVCLASS Command

The QUERY DEVCLASS command can be used to display information about device classes defined with *devtype=server*:

```
adsm> query devclass targetclass

Device        Device         Storage     Device       Format      Est/Max      Mount
Class         Access            Pool     Type                    Capacity      Limit
Name          Strategy         Count                                 (MB)
-----------   ----------     -------     ---------     ------     --------     ------
TARGETCLASS   Sequential           1     SERVER                     1024.0          5
```

Once you have performed these three steps, server-to-server virtual volumes are enabled, and the source server can use the target server to store data.



Database Backup

1. Source server is node of target server
2. Target server defined on source server
3. Device class points to target server

```
adsm> Backup db dev=targetclass type=full
```

Source data stored in filespace on target server as archived objects

Copyright IBM Corporation 1997, 1998

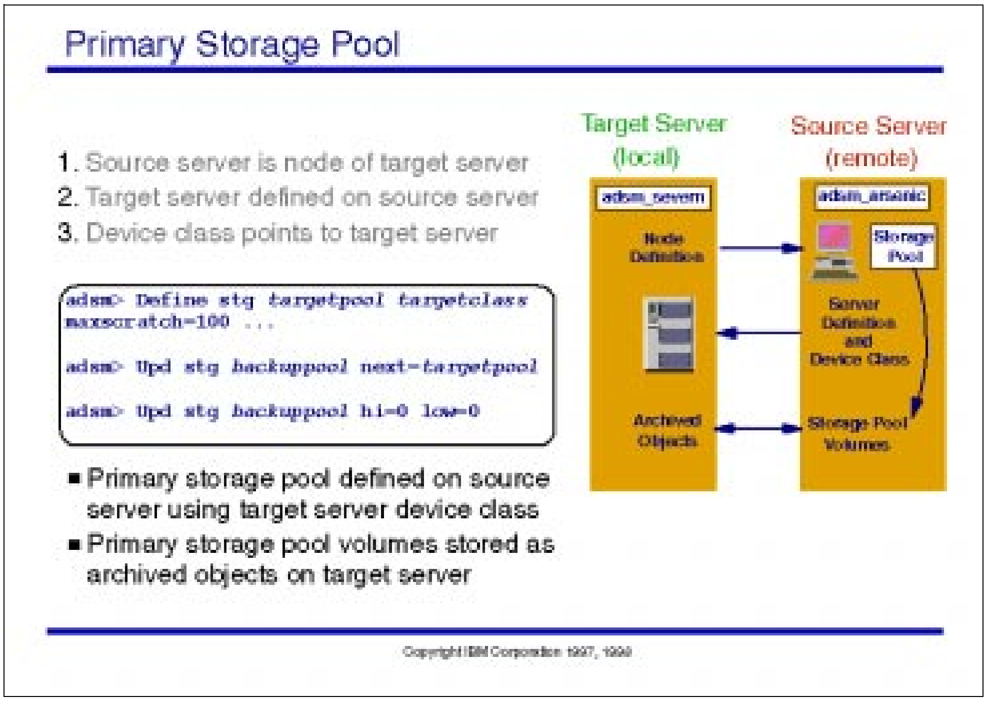
8.1.5 Database Backup

In this example the device class *type=server* can be used to back up the source server database. The database backup is stored as a number of objects on the target server. The number of objects is determined by the *maxcapacity* value of the device class used.

The output from the database backup is stored in volumes, with size determined by the *maxcapacity* parameter of the device class. These appear as database backup volumes on the source server and as archived objects on the target server. The archived objects on the target server are contained in a filespace that belongs to the source server node.

Exports

A device class *type=server* can also be used for EXPORT commands.



Primary Storage Pool

1. Source server is node of target server
2. Target server defined on source server
3. Device class points to target server

```
adsm> Define stg targetpool targetclass
maxscratch=100 ...

adsm> Upd stg backuppool next=targetpool

adsm> Upd stg backuppool hi=0 low=0
```

- Primary storage pool defined on source server using target server device class
- Primary storage pool volumes stored as archived objects on target server

Copyright IBM Corporation 1997, 1999

8.1.6 Primary Storage Pool

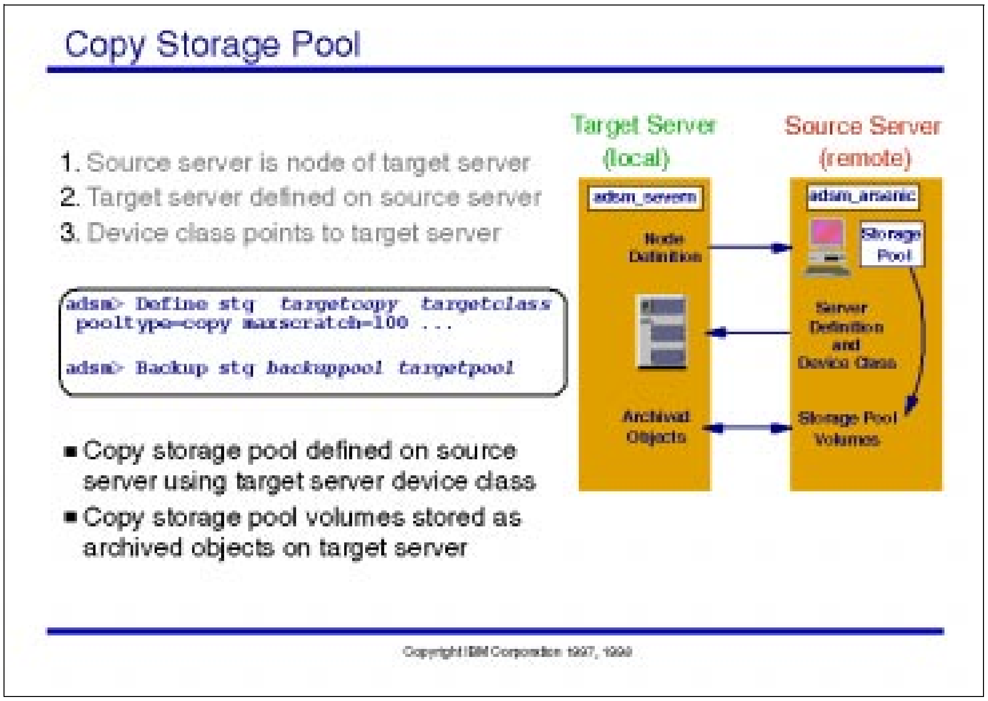
In this example the device class type=server is used to define a primary storage pool within the source server storage hierarchy. Migration of data to this storage pool is started on the source server by lowering the thresholds of the storage pool that migrates to this newly defined storage pool.

8.1.6.1 Primary Storage Pool Definition

The primary storage pool is defined in the usual way, referencing a device class. In this case, the device class points to the target server.

8.1.6.2 Primary Storage Pool Volumes

As data is migrated from the backupool on the source server, the data is stored in volumes in the next storage pool (also on the source server). These volumes are actually created as archived objects on the target server. The size of these volumes is determined by the *maxcapacity* parameter of the device class. Again, these appear as storage pools volumes on the source server and as archived objects on the target server. The archived objects on the target server are contained in a filespace that belongs to the source server node.



Copy Storage Pool

1. Source server is node of target server
2. Target server defined on source server
3. Device class points to target server

```
adsm> Define stg  targetcopy  targetclass
 pooltype=copy maxscratch=100 ...

adsm> Backup stg backuppool targetpool
```

- Copy storage pool defined on source server using target server device class
- Copy storage pool volumes stored as archived objects on target server

Target Server (local)

Source Server (remote)

8.1.7 Copy Storage Pool

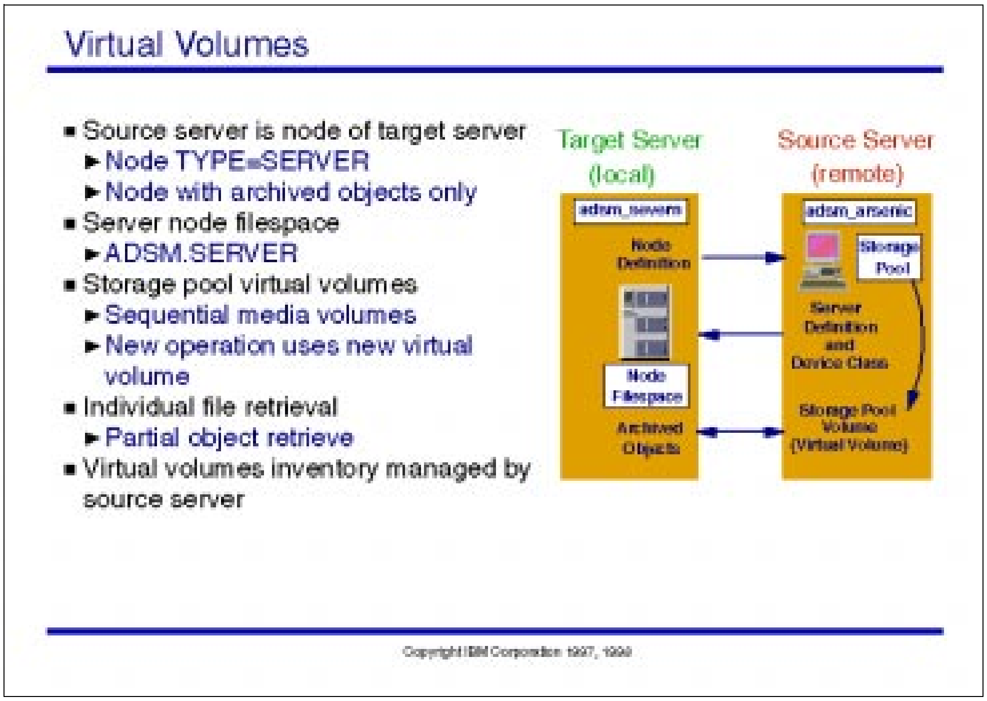
In this final example the device class is used to define a copy storage pool and a backup of an existing primary storage pool to this new copy storage pool is started.

8.1.7.1 Copy Storage Pool Definition

The copy storage pool is defined in the usual way, referencing a device class. In this case, the device class is the target server.

8.1.7.2 Copy Storage Pool Volumes

As data is backed up from the backupool on the source server, it is stored in volumes in the copy storage pool, also on the source server. These volumes are actually created as archived objects on the target server. The size of the volume is determined by the *maxcapacity* parameter of the device class. Again, these appear as copy storage pool volumes on the source server and as archived objects on the target server. The archived objects on the target server are contained in a filespace that belongs to the source server node.



Virtual Volumes

- Source server is node of target server
  - Node TYPE=SERVER
  - Node with archived objects only
- Server node filespace
  - ADSM.SERVER
- Storage pool virtual volumes
  - Sequential media volumes
  - New operation uses new virtual volume
- Individual file retrieval
  - Partial object retrieve
- Virtual volumes inventory managed by source server

8.1.8 Virtual Volumes

Data created on the source server and stored on the target server consists of virtual volumes. These virtual volumes are logically part of a storage pool or database backup series on the source server, but they are physically held as archived objects on the target server.

8.1.8.1 Source Server Is Node of Target Server

The source server is defined as a node on the target server, using the *type=server* option to differentiate it from client nodes. The target server is defined on the source server with the DEFINE SERVER command. These two definitions, and a device class defined on the source server with the *type=server* option, are the only definitions on the source and target that define the relationship between the servers.

Node Types

It is not possible for a node to be both *type=client* and *type=server.* For the same machine to be both types of nodes, different node names must be used.

Nodes defined as *type=server* can only create archived objects on the target server. These archived objects are contained in a filespace on the target server that belongs to the source server node.

8.1.8.2 Server Node Filespace

The filespace on the target server that belongs to the source server node is always called ADSM.SERVER. It has a filespace type of ADSM_FS to differentiate it from other client filespace types, such as NTFS (Windows NT), JFS (AIX), or DB2 (DB2). This filespace can contain multiple storage pool volumes, database backups, or exports from the source server.

8.1.8.3 Storage Pool Virtual Volumes

Storage pools defined on the source server with a *devtype=server* device class contain volumes. These volumes on the source server appear to an administrator just like volumes created with any other device classes. However, these volumes on the source server are not real, they are virtual.

Virtual volumes are used in the same way as any other sequential media. However, it is not possible to append to a virtual volume. When a virtual volume is created, it is not rewritten to. Virtual volumes always have a volume status of "Full" on the source server. They are defined with the device class defined on the source server. They appear on the source server in a format similar to those volumes created with a *devtype=file* device class. Issuing the QUERY VOLUME command on the source server displays all volumes, both real and virtual:

```
adsm> query volume

Volume Name                Storage       Device       Estimated      Pct     Volume
                           Pool Name     Class Name    Capacity      Util     Status
                                                           (MB)
------------------------   -----------   ----------   ---------     -----    --------
D:\ADSM\SERVER\ARC01.DSM   ARCHIVEPOOL   DISK             100.0       0.1    On-Line
D:\ADSM\SERVER\BAK01.DSM   BACKUPPOOL    DISK             100.0      77.3    On-Line
TARGET.BFS.870897083      BACKUPTAPE    TARGETCLASS       20.0     100.0     Full
```

The first two volumes shown above are real volumes in storage pools. The third volume, highlighted, is a scratch virtual volume. The virtual volumes are scratch volumes if the *maxscratch* parameter of the storage pool is greater than zero. In this case, the names of the scratch volumes created take the form: **SERVER.ABC.123456789** where:

SERVER Is the target server name as defined on the source server.

ABC Can be three letters that denote the type of data stored:

- BFS Storage pool data
- EXP Export data
- DBB Database backup
- DMP Database dump

123456789 Nine numeric digits used to give the virtual volume a unique name.

It is also possible to create private virtual volumes. Use the DEFINE VOLUME command and specify a volume name:

```
adsm> define volume backupool tims.new.volume
ANR2206I Volume TIMS.NEW.VOLUME defined in storage pool BACKUPTAPE (device class TARGETCLASS).

adsm> query volume

Volume Name               Storage       Device        Estimated      Pct     Volume
                          Pool Name     Class Name     Capacity     Util     Status
                                                           (MB)
------------------------  -----------   ----------    ---------    -----    --------
D:\ADSM\SERVER\ARC01.DSM  ARCHIVEPOOL   DISK              100.0      0.1    On-Line
D:\ADSM\SERVER\BAK01.DSM  BACKUPPOOL    DISK              100.0     77.3    On-Line
TARGET.BFS.870897083      BACKUPTAPE    TARGETCLASS        20.0    100.0      Full
TIMS.NEW.VOLUME           BACKUPTAPE    TARGETCLASS         0.0      0.0     Empty
```

Private volumes are created empty and are used when required.

The size of these virtual volumes is the maximum of the *maxcapacity* value associated with the device class and the size of the data operation. For example, if the data operation is more than twice the size of the *maxcapacity* value, three volumes are created, and these appear as three archived objects in the ADSM.SERVER filespace on the target server.

The archived objects that correspond to the virtual volumes of the source server are held in a storage pool on the target server. This storage pool on the target server is determined by policy definitions on the target server. It is defined in the archive copy group *destination* parameter, of the default management class, of the policy domain in which the source server node is registered.

8.1.8.4 Individual File Retrieval

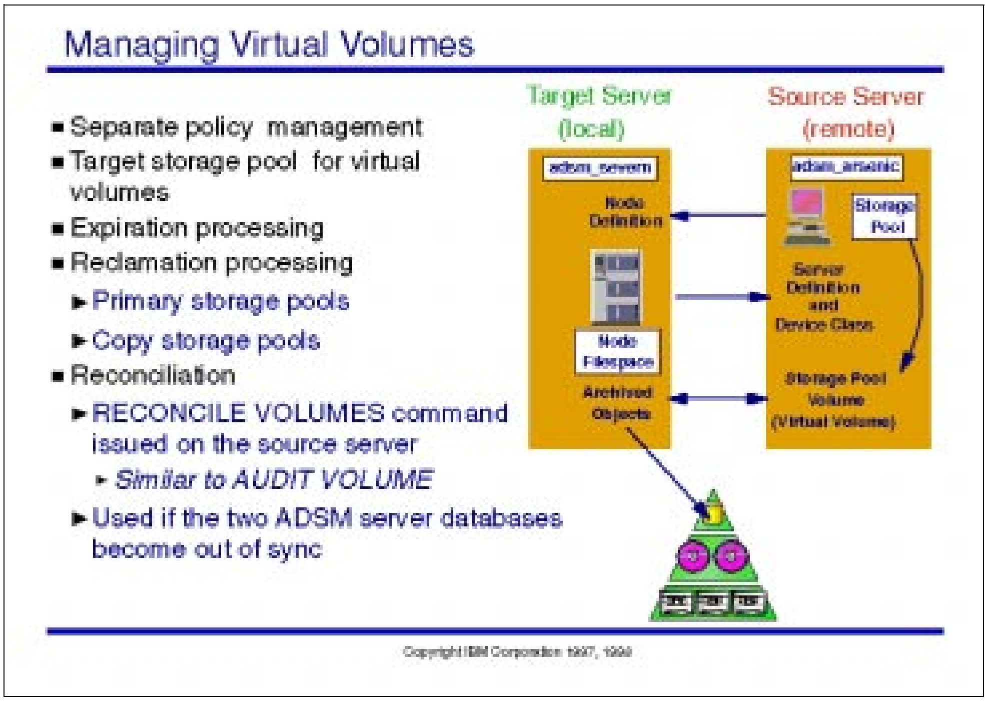
An ADSM client can restore or retrieve as few files from the source server as needed, even when those files are held in the storage pool located on the target server. When files are held in the storage pool on the target server, the source server uses the Partial Object Retrieve mechanism to retrieve only those files from the storage pool, not the entire storage pool.

Although the ADSM client may see the data as either backed up or archived to the source server, the data is always archived from the source server to the target server.

8.1.8.5 Virtual Volume Inventory Managed by Source Server

The target server sees the virtual volumes as archived objects. If these objects were managed by the target server's policy management, as with regular client archive files, only one management class and copy group would be associated with each object (individual virtual volumes from the source server). Given that the virtual volumes make up a storage pool, it would only be possible to have one management class associated with any one storage pool. This would be a severe limitation and would increase the number of storage pools required.

What is needed is the ability to keep multiple files in the same storage pool, such that each file can have a different management class. This is consistent with ADSM Version 2 and can only be done if the source server controls the management of these files, because only the source server has database entries for each file. The target server has only one database entry for the entire storage pool volume.



Managing Virtual Volumes

- Separate policy management
- Target storage pool for virtual volumes
- Expiration processing
- Reclamation processing
  - Primary storage pools
  - Copy storage pools
- Reconciliation
  - RECONCILE VOLUMES command issued on the source server
    - *Similar to AUDIT VOLUME*
  - Used if the two ADSM server databases become out of sync

Copyright IBM Corporation 1997, 1998

8.1.9 Managing Virtual Volumes

The management of virtual volumes is performed from the source server. The only intervention possible from the target server is for an administrator to delete the entire source server filespace:

```
adsm>delete filespace source ADSM.SERVER type=server
```

where "source" is the name of the source server node. If the type=server parameter is not specified, an administrator cannot delete a server node filespace.

8.1.9.1 Separate Policy Management

The policy management of the source server and target server are separate, both unaware of the other’s existence. To ensure that the source server manages the files held in virtual volumes, the target server’s policy management is disabled for server nodes. The backup copy group attributes and HSM attributes in the management class are never used for server nodes because server nodes can only have archived files.

All attributes of the archive copy group on the target server are ignored, with the exception of the *destination* parameter, which specifies the target storage pool.

8.1.9.2 Target Storage Pool for Virtual Volumes

The target storage pool is the storage pool on the target server where the source server’s virtual volumes will be held. It is determined by the archive copy group of the default management class in the policy domain in which the source server is registered. It may be desirable to have a policy domain on the target server for server nodes, separate from the policy domain for regular client nodes, to ease administration.

Although remote server storage is seen as a single storage pool and device class, it may in fact be several storage pools. For example, the target server may accept the data from the source server into the target storage pool and later migrate the data to a different storage pool. The source server is unaware of this migration, just as an ADSM client is unaware of the location of its data.

8.1.9.3 Expiration

The source server controls the expiration of virtual volume data. A target server cannot delete virtual volume information until requested to do so by the source server.

The source server retains individual files in the storage pool for the period specified by their policy management. After the specified period, the source server deletes its database entries.

8.1.9.4 Reclamation

Storage pools containing virtual volumes have reclamation options defined in the same way as other storage pools. When a virtual volume contains expired data, it is a candidate for reclamation, which begins when a threshold is reached.

- Primary storage pool virtual volumes

  For primary storage pool virtual volumes, data being reclaimed on the volume is transferred from the target server to the source server and back to the target server into a new virtual volume.

  If the data is still in the target storage pool on the target server, two mount points are needed in that storage pool. However, the target server may have migrated the data from the target storage pool down its own storage hierarchy. In this case, one mount point is needed in the storage pool where the data now resides, and one mount point is needed in the target storage pool where the new virtual volume will be created with the reclaimed data.

  When reclamation has completed, the reclaimed virtual volume is marked empty and ready for deletion. The source server tells the target server to delete the volume. This is simply a DELETE ARCHIVE operation, because the virtual volumes are kept as archived objects on the target server. On receiving the DELETE ARCHIVE request, the target server waits for a further period, specified by the *delgraceperiod* attribute of the DEFINE SERVER command on the source server, before deleting the reclaimed virtual volume.

- Copy storage pool virtual volumes

  For copy storage pool virtual volumes, reclaimed data is copied from the primary storage pool wherever possible. For copy storage pool virtual volumes, this copy operation minimizes the network traffic between the two servers to a single session, assuming that the primary volume is on the source server. When the data is written to a new copy storage pool virtual volume, the old copy storage pool virtual volumes are deleted by the source server. The source server tells the target server to delete the reclaimed virtual volumes after the *reusedelay* time delay has passed, to ensure that the server database and server storage can be restored to a previous point in time if required.

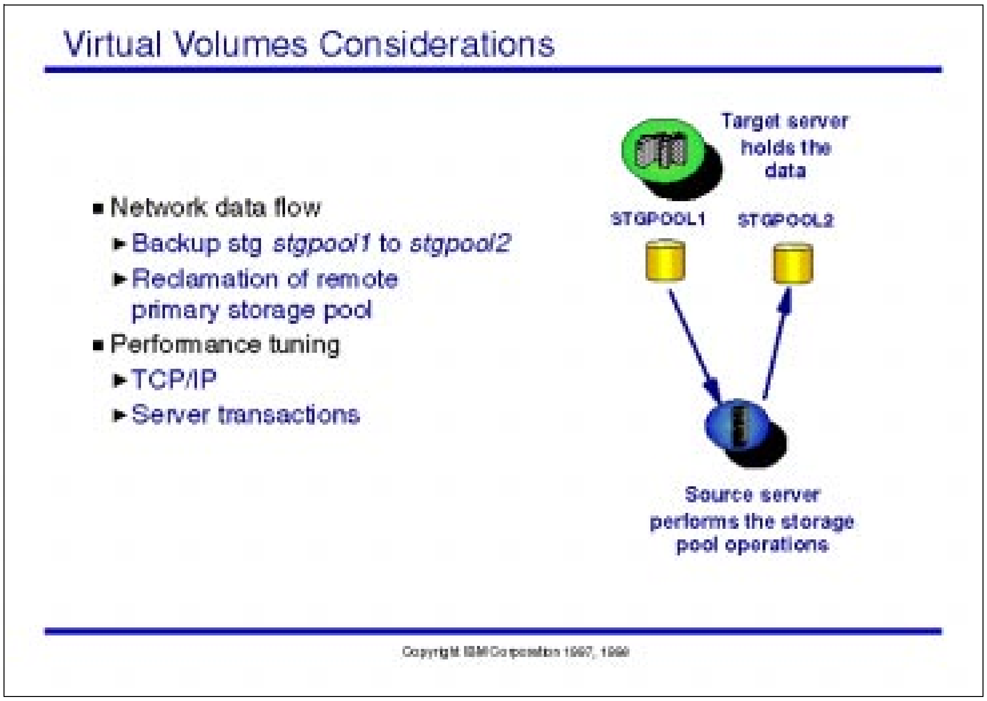
On receiving the DELETE ARCHIVE request, the target server waits for a further delay, as specified by the *delgraceperiod* attribute associated with the definition of the target server at the source server.

8.1.9.5 Reconciliation

Reconciliation can be performed through the new RECONCILE VOLUMES command when differences in the virtual volume definitions on the source and target servers have occurred. For example:

- The target server was temporarily unavailable when the source server ran an expiration process on data that resides on the target server.
- One of the servers was subject to a disaster and a database restore had to be done to an earlier point in time.
- A DELETE VOLHIST command was issued on the source server at a time when the target server was unavailable.
- A virtual volume definition was deleted from the source server at a time when the target server was unavailable.

The RECONCILE VOLUMES command can be issued from the source server to reconcile differences in the virtual volume definitions on the source server and archived objects on the target server. Both servers must be running to use this command.



Virtual Volumes Considerations

- Network data flow
  - Backup stg *stgpool1* to *stgpool2*
  - Reclamation of remote primary storage pool
- Performance tuning
  - TCP/IP
  - Server transactions

Target server holds the data
STGPOOL1
STGPOOL2
Source server performs the storage pool operations

8.1.10 Virtual Volumes Considerations

There are considerations for managing server-to-server virtual volumes.

8.1.10.1 Network Data Flow

ADSM server-to-server virtual volumes duplicates data over the network when:

- A storage pool is backed up to a copy storage pool where both storage pools are defined on the same source server and held on the same target server.

  In this situation, the source server performs the storage pool backup operation. The data is transferred from the primary storage pool volume, which resides on the target server, to the source server, and then back to a copy storage pool volume which also resides on the target server.

- The primary storage pool held on a target server is reclaimed.

  In this situation, data is moved from one primary storage pool volume (which resides on the target server) to another primary storage pool volume (which also resides on the target server).

8.1.10.2 Performance Tuning

Two types of performance tuning can be done to optimize ADSM server-to-server virtual volumes:

- TCP/IP

  TCP/IP system parameters such as *tcp_sendspace* and *tcp_recvspace* (examples from AIX) can be optimized with values that depend on the network topology being used. These parameters must be optimized at both ADSM servers.

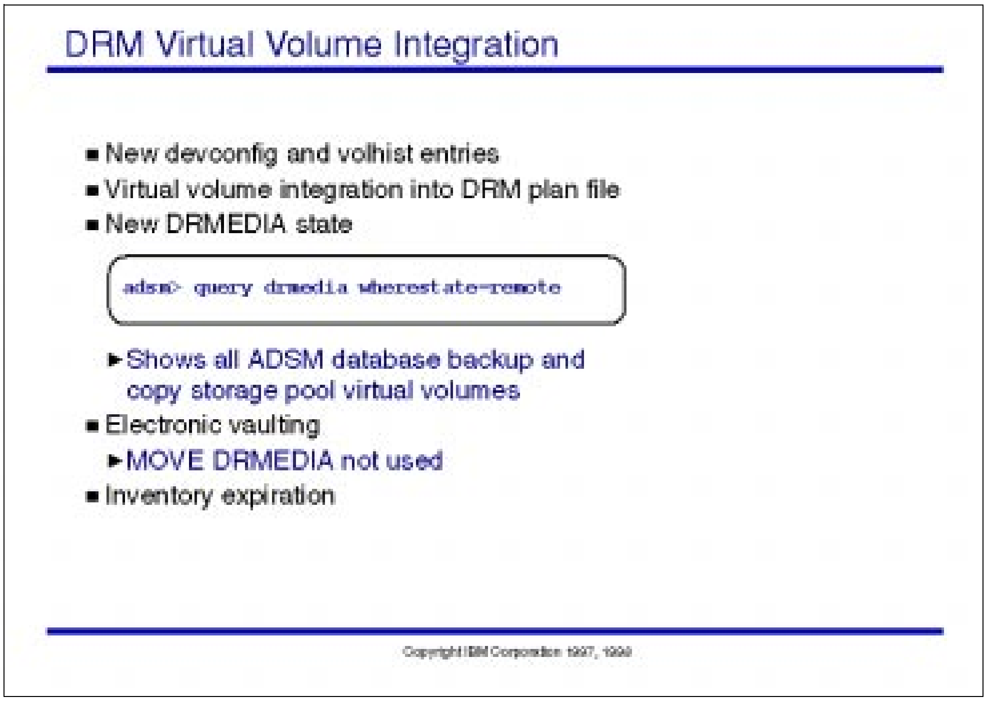
  TCP/IP parameters in the target server option file can be tuned, namely, *tcpwindowsize* and *tcpnodelay.*

The configuration of the network adapter device used in each ADSM server can be optimized.

- Server transactions

  All server data is moved in transactions controlled by the *movebatchsize* and *movesizethresh* server options on the source server. *movebatchsize* specifies the number of files that are to be moved within the same server transaction, and *movesizethresh* specifies, in megabytes, the amount of data to be moved within the same server transaction. When either threshold is reached, a new transaction is started.

  The target server treats data movement to or from the source server in the same way as for client nodes. The *txngroupmax* parameter in the server options file determines when a new node transaction is started. For best performance, this transaction size should be the same as the source server transaction size determined by the *movebatchsize* and *movesizethresh* server options.



DRM Virtual Volume Integration

- New devconfig and volhist entries
- Virtual volume integration into DRM plan file
- New DRMEDIA state

```
adsm> query drmedia wherestate=remote
```

  - Shows all ADSM database backup and copy storage pool virtual volumes
- Electronic vaulting
  - MOVE DRMEDIA not used
- Inventory expiration

Copyright IBM Corporation 1997, 1999

8.1.11 DRM Virtual Volume Integration

There are several factors to consider when implementing DRM with server-to-server virtual volumes.

8.1.11.1 New Devconfig and Volhist Files

A device class that uses the target server requires that the server must first be defined on the ADSM source server. To support this requirement, the DEFINE SERVER command is included in the device configuration file.

Virtual volumes are tracked in the source server volume history file in the same way as physical volumes.

8.1.11.2 Virtual Volume Integration into DRM Plan File

The DRM stanzas that have been updated to work with virtual volumes are:

- RECOVERY.VOLUMES.REQUIRED
  - List of volumes required for database restore. Includes backup virtual volumes if they are part of the latest backup series.
  - Volumes required for storage pool restore will include copy storage pool virtual volumes.

  Volumes in this stanza are grouped according to location. For virtual volumes, the location field contains the server name.
- RECOVERY.DEVICES.REQUIRED

  This stanza provides details about the devices required to read the volumes included in the RECOVERY.VOLUMES.REQUIRED stanza. Details include device class definitions associated with DEVTYPE=SERVER.
- DEVICE.CONFIGURATION.FILE

This stanza includes administrative server table definitions and device class definitions that correspond to virtual volumes.

- LICENSE.INFORMATION

  This stanza displays the license information associated with virtual volumes.

- LICENSE.REGISTRATION

  For server platforms such as Windows NT that use the license certificate method, this stanza includes the register license command required to enable the virtual volume function.

Stanzas that have not been updated but work with virtual volumes are:

- SERVER.REQUIREMENTS
- SERVER.INSTRUCTIONS.xxxx
- RECOVERY.SCRIPT.DISASTER.RECOVERY.MODE
- RECOVERY.SCRIPT.NORMAL.MODE
- Create/format/install/restore DB start server stanzas
- COPYSTGPOOL.VOLUMES.AVAILABLE

  Contains an ADSM macro with the commands used to mark copy storage volumes that were moved offsite as moved back onsite. This stanza does not include copy storage pool virtual volumes because their access does not have to be updated after a database is restored.

- COPYSTGPOOL.VOLUMES.DESTROYED

  Contains an ADSM macro with the commands used to mark copy storage volumes as unavailable that were onsite at the time of the disaster. This stanza does not include copy storage pool virtual volumes because these volumes are considered offsite and should not be marked as destroyed.

- PRIMARY.VOLUMES.DESTROYED

  Contains an ADSM macro with the commands used to mark primary storage volumes as destroyed. This stanza does not include primary virtual volumes because these volumes are considered offsite and should not be marked as destroyed.

- PRIMARY.VOLUMES.REPLACEMENT.CREATE

  Contains a shell script with the commands used to re-create primary storage volumes. PREPARE assumes that the primary virtual volumes have not been destroyed.

- PRIMARY.VOLUMES.REPLACEMENT

  Contains an ADSM macro with the commands used to define replacement primary storage volumes to the ADSM server. PREPARE assumes the primary virtual volumes have not been destroyed.

8.1.11.3 New DRMEDIA State

The QUERY DRMEDIA command has been updated for the new REMOTE state introduced for virtual volumes. Issuing the QUERY DRMEDIA WHERESTATE=REMOTE command displays:

- Database backup virtual volumes
- Copy storage pool virtual volumes

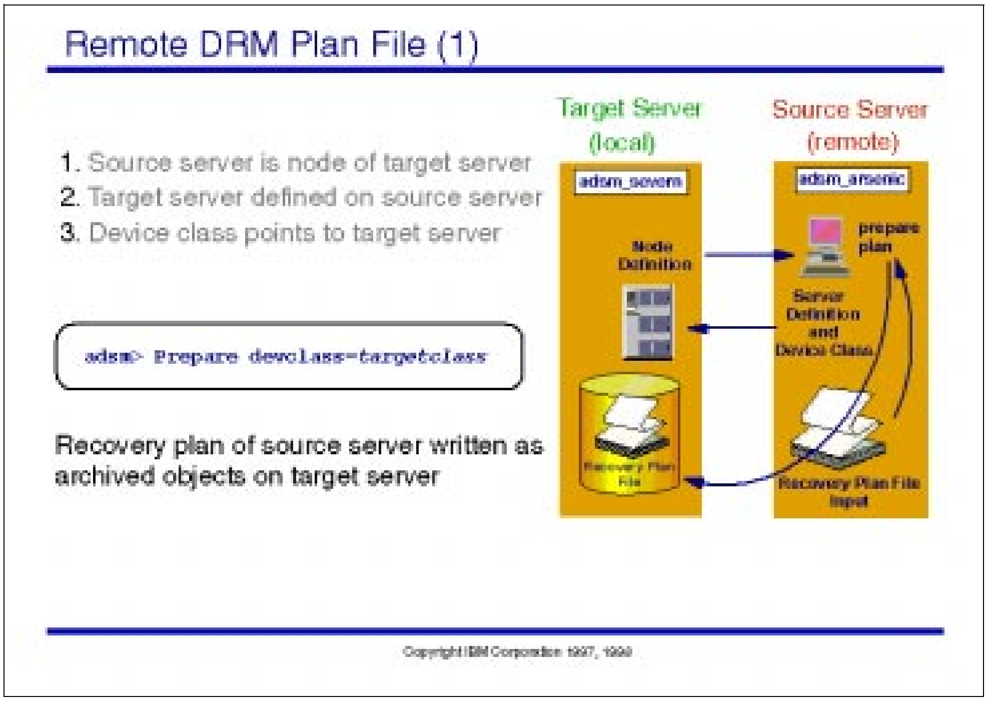
Because a virtual volume location contains the server name, the WHERELOCATION parameter can be used to display all volumes associated with a specific server.

8.1.11.4 Electronic Vaulting

For copy storage pool volumes with Version 2 that were removable media, the MOVE DRMEDIA command is used to change the location status of the volumes in the ADSM database. However, virtual volumes are not moved, because a form of electronic vaulting is used. The MOVE DRMEDIA command is not used.

8.1.11.5 Inventory Expiration

If DRM is licensed, the inventory expiration process will expire a virtual volume database backup series according to the SET DRMDBBACKUPEXPIREDAYS parameter. Thus expiration is automatically controlled.



Remote DRM Plan File (1)

1. Source server is node of target server
2. Target server defined on source server
3. Device class points to target server

```
adsm> Prepare devclass=targetclass
```

Recovery plan of source server written as archived objects on target server

Copyright IBM Corporation 1997, 1998

8.1.12 Remote DRM Plan File (1)

DRM has been enhanced to make use of server-to-server virtual volumes for storing the DRM plan file on a remote server. The following terms are used to differentiate between the two servers shown in this configuration:

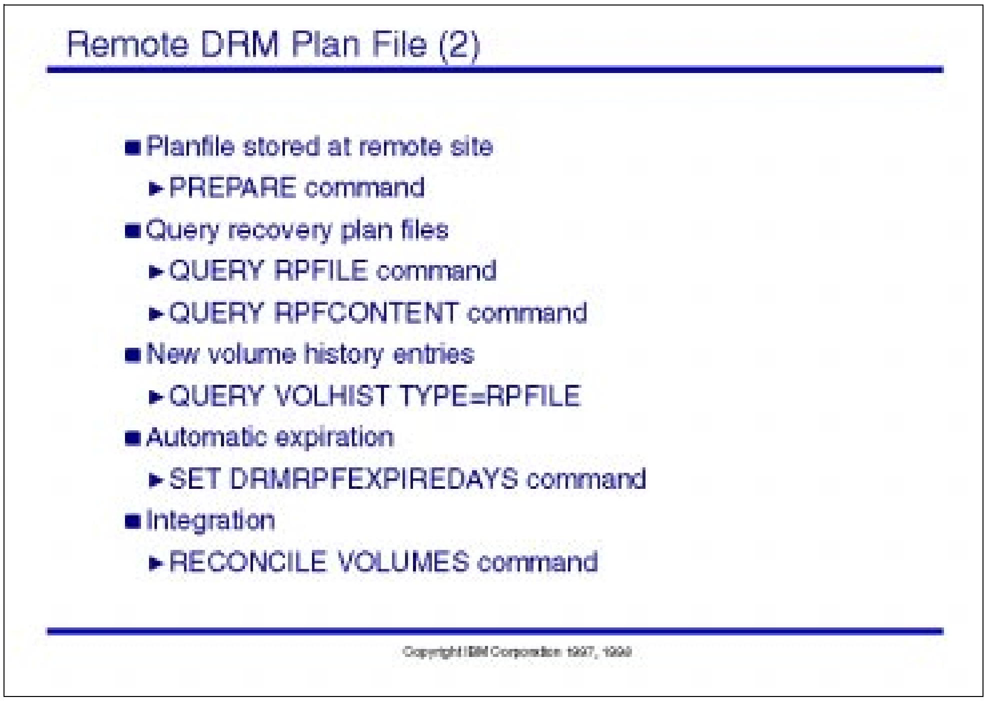
- *Source server* - refers to the server that runs DRM to prepare for disaster recovery. The source server creates the DRM plan files.
- *Target server* - refers to the remote server that is the server which stores the DRM plan files of the target server

Because remote DRM plan files use server-to-server virtual volumes, the setup described in “Configuration (1)” on page 244 through “Configuration (3)” on page 248 must be defined first.

With this preparation the source server can take advantage of the new DEVCLASS parameter for the PREPARE command:

```
adsm> prepare devclass=targetclass
```

This command writes the recovery plan file to the repository pointed to by the device class *targetclass*. This repository is the storage pool that is the target of the archive copy group for the default management class used for the node definition for the source server on the target server.



Remote DRM Plan File (2)

- Planfile stored at remote site
  - PREPARE command
- Query recovery plan files
  - QUERY RPFILE command
  - QUERY RPFCONTENT command
- New volume history entries
  - QUERY VOLHIST TYPE=RPFILE
- Automatic expiration
  - SET DRMRPFEXPIREDAYS command
- Integration
  - RECONCILE VOLUMES command

Copyright IBM Corporation 1997, 1999

8.1.13 Remote DRM Plan File (2)

This graphic lists new commands for remote DRM plan files and other changes that come with this feature.

8.1.13.1 Plan File Stored at Remote Site

The PREPARE command generates a recovery plan file. The recovery plan file contains the information and procedures that are needed to recover an ADSM server to the most recent completed backup operation. The PREPARE command has a new DEVCLASS parameter:

- DEVclass=devclassname

  Specifies the device class used to create a recovery plan file object on the target server. The device class must have a device type of SERVER. This parameter is optional. When this parameter is specified, the recovery plan file is written as an object on the target server, and a record is created in the volume history table on the source server.

  The maximum capacity specification for the DEVCLASS attribute must be larger than the size of the recovery plan file. If the size of the recovery plan file exceeds the maximum capacity, the PREPARE command fails.

  The recovery plan file archive object naming convention on the target server is:

  Filespace name: ADSM.SERVER
  High-level qualifier: devclassprefix\servername.yyyymmdd.hhmmss
  Low-level qualifier: RPF.OBJ.1

  The recovery plan file virtual volume name recorded in the volume history table on the source server is:

  servername.yyyymmdd.hhmmss

8.1.13.2 Query Recovery Plan Files

Two new QUERY commands are introduced to display remote DRM plan files.

The QUERY RPFILE command applies to recovery plan files created by using the DEVCLASS parameter with the PREPARE command. The command displays a list of recovery plan files. A list of recovery plan files can be displayed by querying either the server that created the recovery plan files (the source server) or the server that was used to store the recovery plan files (the target server). Either DEVCLASS or NODENAME has to be specified.

When the DEVCLASS parameter is specified, these considerations apply:

- A list of the recovery plan files that were created for this server is displayed.
- This command is issued to the same server that ran the PREPARE command (the source server).
- The specified device class name was used on the PREPARE command that created the recovery plan file.

When the NODENAME parameter is specified, these considerations apply:

- A list of the recovery plan files that are stored in this server is displayed.
- This command is issued to the server that was the target of the PREPARE command.
- The specified node name is registered in this server with a node type of SERVER.
- This option is used when the ADSM server that created the recovery plan file is not available:

```
adsm> q rpfile devcl=*

Recovery Plan File Name          Node Name             Device Class Name
-------------------------------  --------------------- --------------------
ADSM_SEVERN.19980722.141747      SERVER_INDUS          SEVERN_CL

ANS5103I Highest return code was 0.
```

The QUERY RPFCONTENT command also only applies to recovery plan files created by using the DEVCLASS parameter with the PREPARE command. The command displays the contents of a specific recovery plan file. This command cannot be issued from the server console. The Web administrative interface also allows you to display the contents of individual or multiple recovery plan file stanzas.

The contents of a recovery plan file can be displayed by querying either the server that created the recovery plan file (the source server) or the server that was used to store the recovery plan file (the target server). The name of the recovery plan file and either the DEVCLASS or NODENAME parameter have to be specified.

When the DEVCLASS parameter is specified, these considerations apply:

- The contents of a recovery plan file that was created for this server are displayed.
- This command is issued to the same server that ran the PREPARE command (the source server).

- The specified device class name was used on the PREPARE command that created the recovery plan file.

When the NODENAME parameter is specified, these considerations apply:

- The contents of a recovery plan file that was stored in this server are displayed.
- This command is issued to the server that was the target of the PREPARE command.
- The specified node name is registered in this server with a node type of SERVER.
- This option is used when the ADSM server that created the recovery plan file is not available:

```
adsm> q rpfc ADSM_SEVERN.19980722.141747 devcl=severn_cl
```

8.1.13.3 New Volume History Entries

A new volume history entry with the following contents is created each time a PREPARE DEVCLASS command is run:

- Volume type: *rpfile*
- Volume name format: *servername.yyyymmdd.hhmmss*
- Server name of target server

```
adsm> query volhist type=rpfile

       Date/Time: 07/22/1998 14:17:47
     Volume Type: RPFILE
   Backup Series: 0
Backup Operation: 0
      Volume Seq: 1
    Device Class: SEVERN_CL
     Volume Name: ADSM_SEVERN.19980722.141747
 Volume Location: ADSM_SEVERN
```

8.1.13.4 Automatic Expiration

A new SET DRMRPFEXPIREDAYS command introduces the ability to automate expiration of remote DRM plan files after the number of days specified by *n*.

This command controls how long to keep the objects that contain the recovery plan files. Expiration processing on the source server automatically expires plan files that are saved on the target server. DRM plan files will expire *n* days after creation. The last remote DRM plan file will not expire. Locally created recovery plan files are not automatically expired.

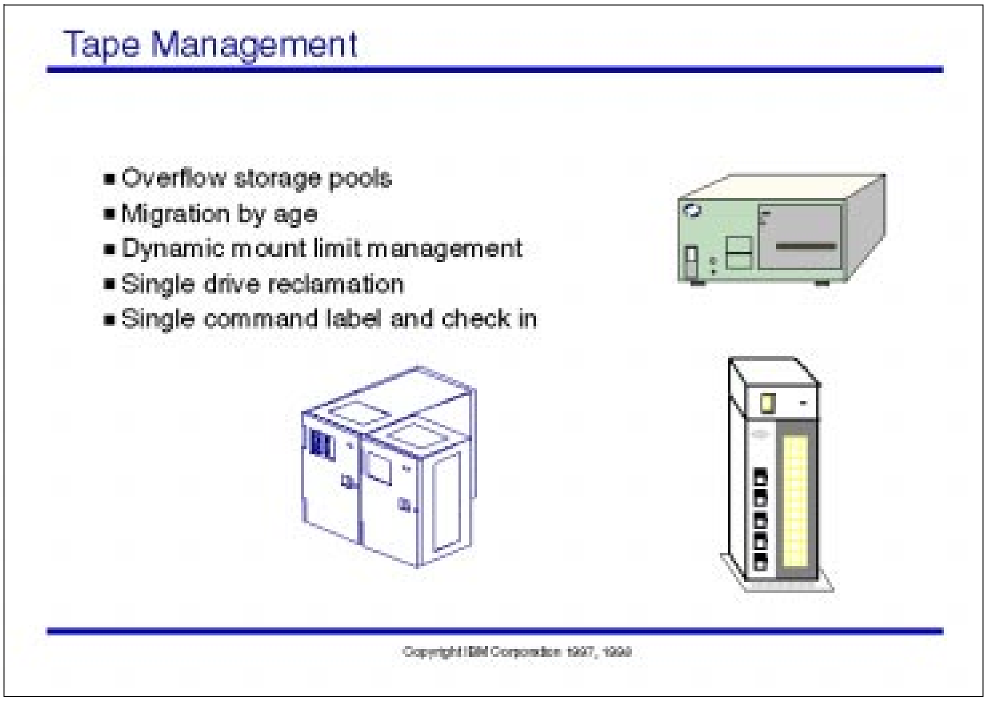
```
07/23/1998 15:27:46  ANR2369I Database backup volume and recovery plan file
                      expiration starting under process 17.
07/23/1998 15:27:46  ANR0812I Inventory file expiration process 17 completed:
                      examined 8 objects, deleting 0 backup objects, 0 archive
                      objects, 0 DB backup volumes, and 1 recovery plan files.
                      0 errors were encountered.
```

8.1.13.5 Integration

The RECONCILE VOLUMES command is used to reconcile differences in the virtual volume recovery plan file definitions on the source server and the archived recovery plan file objects on the target server.

When issued on the source server, this command finds all recovery plan file volumes on the source server and all archived recovery plan file objects recorded in the target server inventory that are defined with the specified device class (which must have a device type of SERVER). The target server inventory is also compared to the local definition for recovery plan file volumes to see whether inconsistencies exist. If a connection cannot be established to the target server, the command fails.

8.2 Tape Management



Tape Management

- Overflow storage pools
- Migration by age
- Dynamic mount limit management
- Single drive reclamation
- Single command label and check in

This section describes the enhancements that have been made to the management of tape devices and volumes in ADSM Version 3:

- Overflow Storage Pools

  A new concept introduced to aid administrators in the management of sequential media storage pools that contain more volumes than can be held in a library. Primarily for long-term archive storage pools.

- Dynamic Mount Limit Management

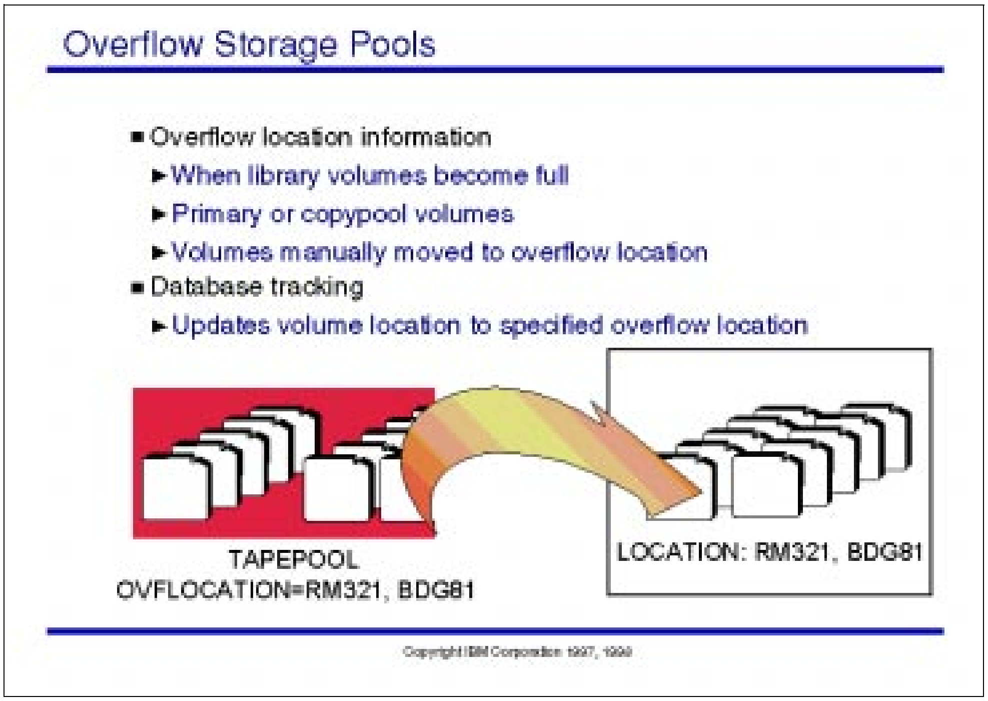
  New options for management of device class mount limits that enable ADSM to monitor the number of available drives and adjust the mount limits accordingly

- Single Drive Reclamation

  Configuration steps necessary to support automated single-drive reclamation in ADSM Version 3

- Single Command Label and Check in

  ADSM Version 3 enhancement to allow DSMLABEL and CHECKIN LIBVOL to be completed in one step while ADSM is running and the drive is defined and on-line to the server



Overflow Storage Pools

- Overflow location information
  - When library volumes become full
  - Primary or copypool volumes
  - Volumes manually moved to overflow location
- Database tracking
  - Updates volume location to specified overflow location

Copyright IBM Corporation 1997, 1998

8.2.1 Overflow Storage Pools

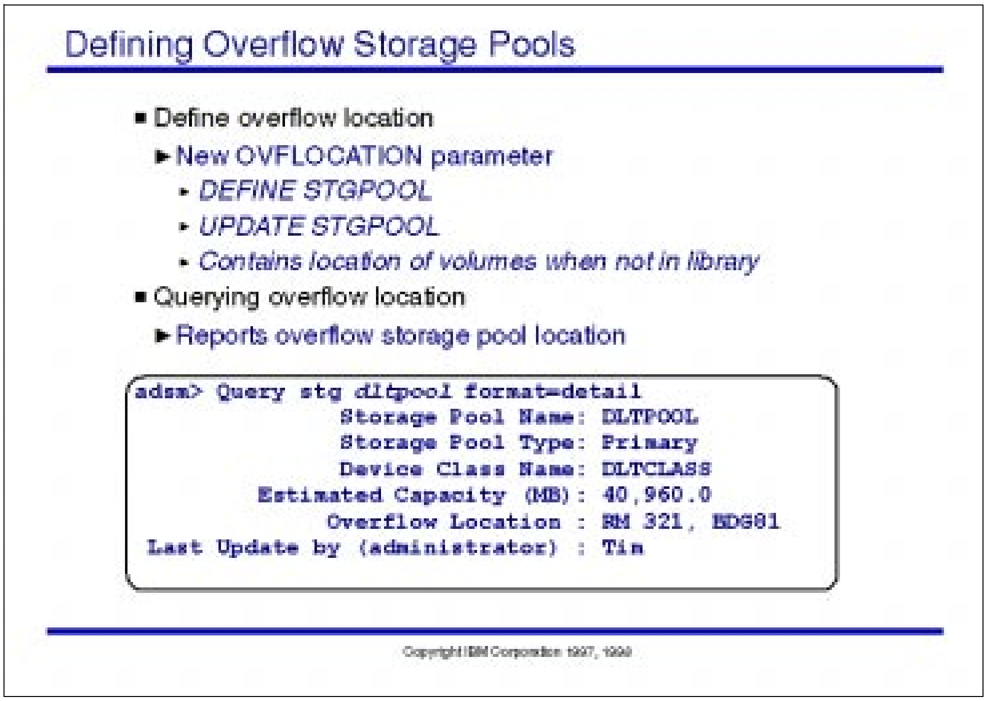
The concept of overflow storage pools has been introduced to allow the ADSM administrator to better manage storage pools that grow beyond the capacity of the library. It is intended to be used primarily for long-term archive storage pools where the data is maintained inside but is rarely retrieved for ADSM processing. Although similar to the DRM media management command, MOVE DRMEDIA, the overflow storage concept is for use only with inside volumes and does not replace the DRM commands.

8.2.1.1 Overflow Location Information

An overflow storage pool can be used for both primary and copy storage pools and allows, when a library becomes full, the removal and tracking of some of the volumes to an overflow location. An overflow storage pool is not a physical storage pool; it is a location where volumes are physically moved to, having been removed from a physical library.

8.2.1.2 Database Tracking

The server database keeps track of volumes that have been moved from libraries to overflow storage pools by updating the individual volume location. When a volume within a library is moved to an overflow location, the volume’s location is updated with that information.



Defining Overflow Storage Pools

- Define overflow location
  - New OVFLOCATION parameter
    - *DEFINE STGPOOL*
    - *UPDATE STGPOOL*
    - *Contains location of volumes when not in library*
- Querying overflow location
  - Reports overflow storage pool location

```
adsm> Query stg dltpool format=detail
                     Storage Pool Name: DLTPOOL
                     Storage Pool Type: Primary
                     Device Class Name: DLTCLASS
             Estimated Capacity (MB): 40,960.0
                    Overflow Location : RM 321, BDG81
 Last Update by (administrator) : Tim
```

Copyright IBM Corporation 1997, 1999

8.2.2 Defining Overflow Storage Pools

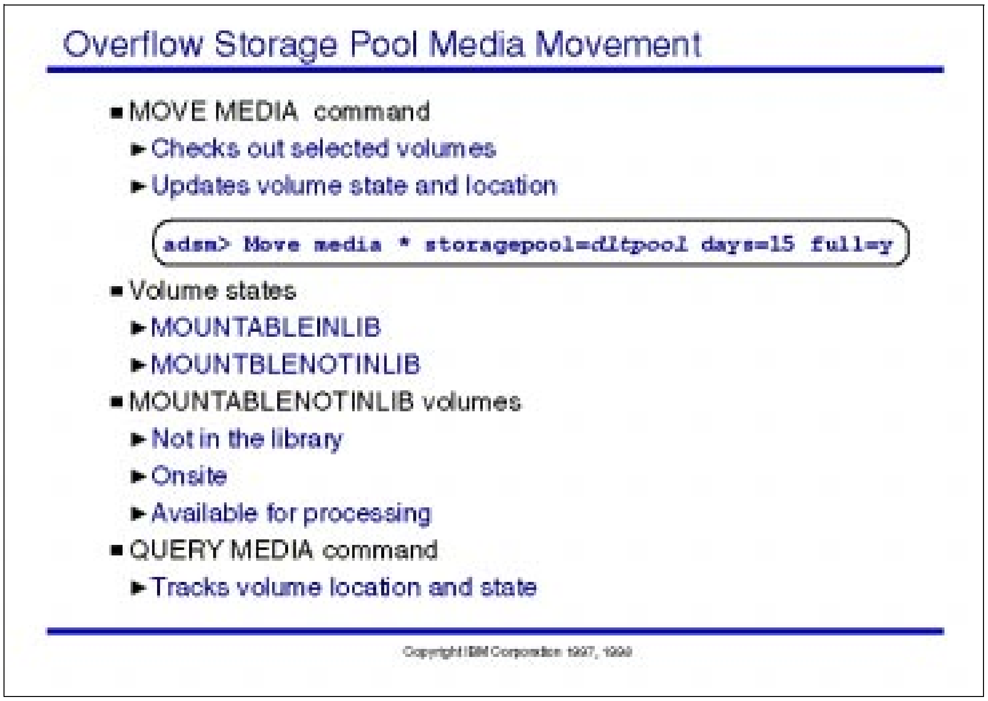
An important aspect of overflow storage pools is the location of storage pool volumes when they are moved from the library.

8.2.2.1 Define Overflow Location

The first stage in implementing an overflow storage pool is to define the location of the volumes when they overflow from the library. Use either the DEFINE STGPOOL or UPDATE STGPOOL command and the new *ovflocation* parameter, which is a text description of where the volumes can be found. There is no default value for *ovflocation*. The description can be up to 255 characters long.

8.2.2.2 Querying Overflow Location

To display the overflow location for a storage pool use the QUERY STG *format=detail* command. As you can see from the sample output, this command shows the storage pool name, type, associated device class, estimated capacity, and overflow location.



Overflow Storage Pool Media Movement

- MOVE MEDIA command
  - Checks out selected volumes
  - Updates volume state and location

```
adsm> Move media * storagepool=dltpool days=15 full=y
```

- Volume states
  - MOUNTABLEINLIB
  - MOUNTBLENOTINLIB
- MOUNTABLENOTINLIB volumes
  - Not in the library
  - Onsite
  - Available for processing
- QUERY MEDIA command
  - Tracks volume location and state

Copyright IBM Corporation 1997, 1998

8.2.3 Overflow Storage Pool Media Movement

ADSM Version 3 introduces some new commands and volume states to manage the movement of storage pool volumes to and from the overflow location.

8.2.3.1 MOVE MEDIA Command

The new MOVE MEDIA command enables the administrator to move volumes from the library to the overflow location. With this command volumes can be selected by name including wildcard values, by the number of days elapsed after the volume has been written or read, by their status (full, filling, empty), by the specified storage pool, and by their state (mountableinlib, nomountableinlib). Volumes processed with the MOVE MEDIA command are checked out of SCSI libraries.

8.2.3.2 Volume States

Two new volume states have been introduced to manage the overflow storage volumes:

- MOUNTABLEINLIB indicates that the storage volume is in the library and available for ADSM processing whenever needed.
- MOUNTABLENOTINLIB indicates that the volume is inside in the overflow storage pool location.

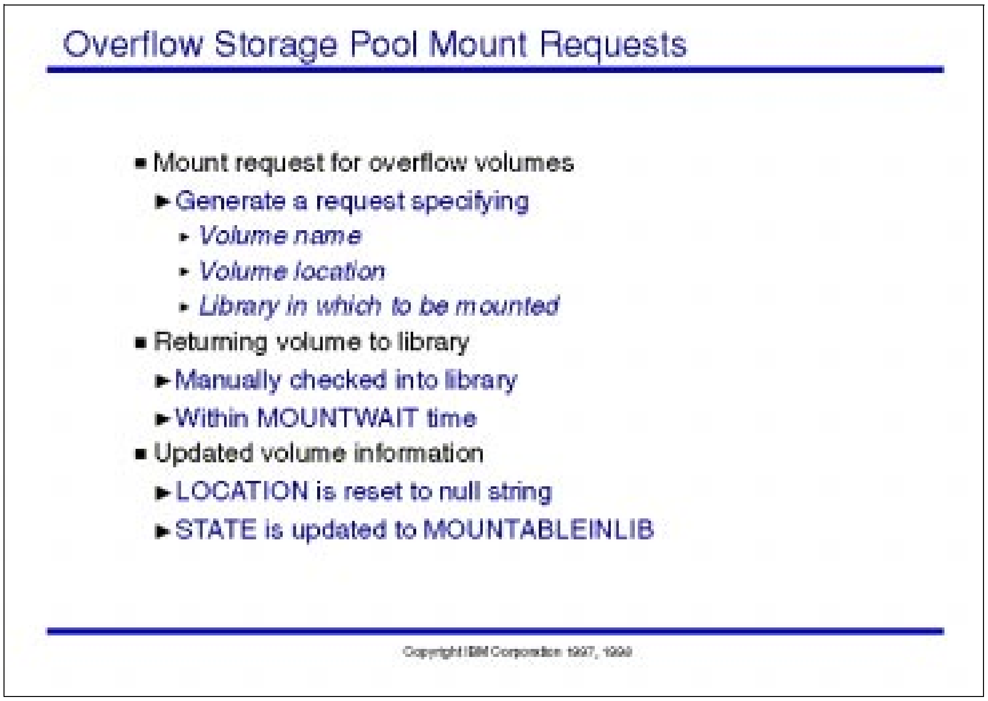
When the MOVE MEDIA command is issued against a number of volumes in a storage pool, an alternative overflow location can be specified. The volumes are then checked out of the library and their status in the ADSM database is updated to the state of *mountablenotinlib*. The location of the volume is also updated to either the storage pool default or the new location specified on the command line.

8.2.3.3 MOUNTABLENOTINLIB Volumes

A volume whose state is *mountablenotinlib* is considered to be inside and available for processing but not in the library. These volumes are not updated as unavailable, and ADSM expects to be able to issue mount requests for them if required.

8.2.3.4 QUERY MEDIA Command

For the tracking of overflow volumes a new QUERY MEDIA command has been introduced. The parameters for this command enables an administrator to query volumes in a storage pool by name, by state, by location, or full volumes only. Thus the tracking of the volumes in any location or state is accurate, and reports can be easily generated for auditing purposes.



Overflow Storage Pool Mount Requests

- Mount request for overflow volumes
  - Generate a request specifying
    - *Volume name*
    - *Volume location*
    - *Library in which to be mounted*
- Returning volume to library
  - Manually checked into library
  - Within MOUNTWAIT time
- Updated volume information
  - LOCATION is reset to null string
  - STATE is updated to MOUNTABLEINLIB

Copyright IBM Corporation 1997, 1998

8.2.4 Overflow Storage Pool Mount Requests

Once the volumes have been successfully moved to the overflow storage pool location, ADSM may subsequently need to access the volumes for processing. To do this it issues a mount request.

8.2.4.1 Mount Request for Overflow Volumes

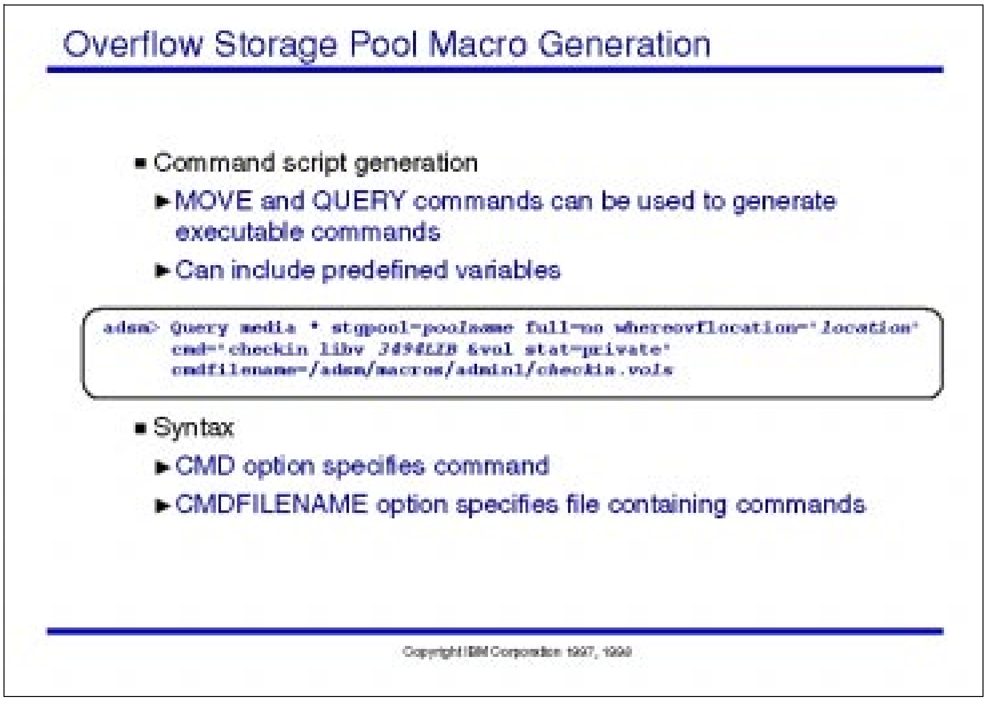
A mount request for a volume in the overflow storage pool location is in the form of a message sent to either the console or the activity log. The message specifies the volume name, volume overflow location, and details of the library in which the volume should be mounted.

8.2.4.2 Returning Volumes to Library

Having issued the mount request, the server then waits the amount of time specified by the *mountlimit* parameter for the volume to be checked in. An administrator must locate the volume and check it into the specified library. If the volume is not returned to the library before the *mountlimit* expires, an error is returned to the requesting operation.

8.2.4.3 Updated Volume Information

If the volume is returned to the specified library before the *mountlimit* expires, the volume location is updated to a null string, and the state is updated to *mountableinlib*. The operation that requested the volume then continues.



Overflow Storage Pool Macro Generation

- Command script generation
  - MOVE and QUERY commands can be used to generate executable commands
  - Can include predefined variables

```
adsm> Query media * stgpool=poolname full=no whereovflocation='location'
      cmd='checkin libv 3494LIB &vol stat=private'
      cmdfilename=/adsm/macros/admin1/checkin.vols
```

- Syntax
  - CMD option specifies command
  - CMDFILENAME option specifies file containing commands

Copyright IBM Corporation 1997, 1999

8.2.5 Overflow Storage Pool Macro Generation

Both the QUERY MEDIA and MOVE MEDIA commands have had additional parameters added to them. The *cmd* and *cmdfilename* parameters allow the creation of script files to further automate overflow storage pool operations.

8.2.5.1 Command Script Generation

To produce a command file from a QUERY or MOVE operation, the *cmd* option is assigned a command string, which may include these predefined variables:

- &VOL - Volume name
- &LOC - Volume location
- &VOLDSN - Volume file name
- &NL - New line character

8.2.5.2 Syntax

The *cmdfilename* parameter specifies the file that is built from the commands produced. This file can then be run by using the MACRO command.

To display full and partially full volumes in the 3590POOL primary storage pool that are stored in overflow location RM321 BDG51 and to generate the CHECKIN LIBVOL command specified with the *cmd* parameter for each volume processed, use the following command:

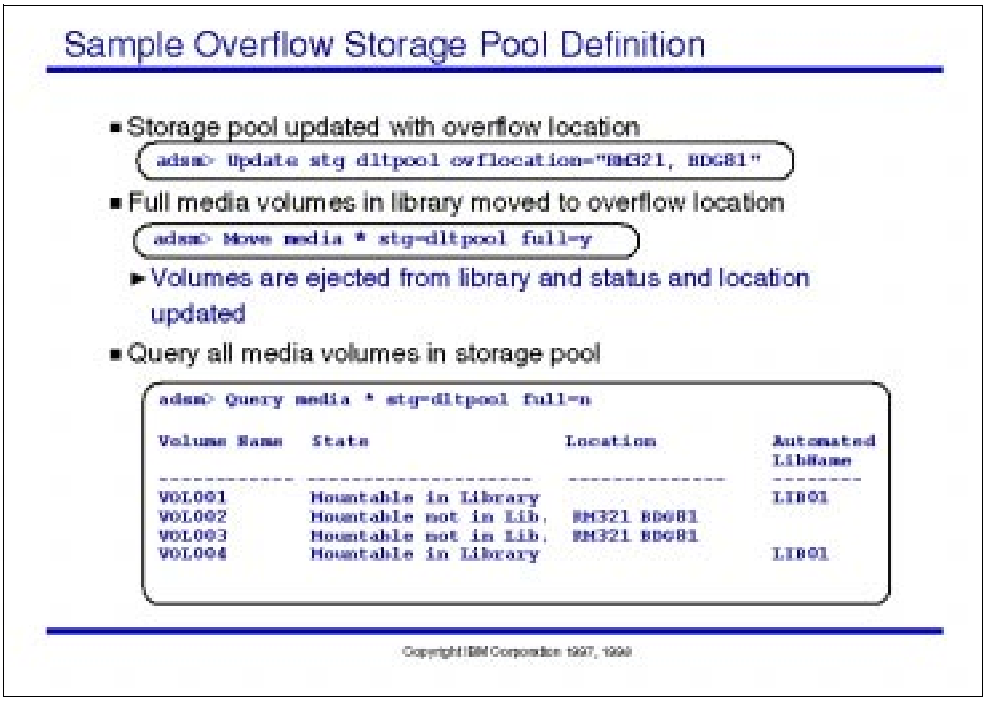
```
adsm>query media * stgpool=3590pool full=no
     whereovflocation="RM321 BDG51"
     cmd="checkin libv 3494LIB &vol stat=private"
     cmdfilename=/adsm/macros/checkin.vols1
```

This command would display the following volumes in the library that match that overflow location:

```
Volume Name   State                   Location                Automated
                                                               LibName
------------  ----------------------  ----------------------  ----------
VOL009        Mountable not in Lib.   RM321 BDG51
VOL012        Mountable not in Lib.   RM321 BDG51
```

It also creates the /adsm/macro/checkin.vols1 file with the matching volumes substituted for the &VOL variable:

```
checkin libovl 3494lib vol009 status=private
checkin libovl 3494lib vol012 status=private
```



Sample Overflow Storage Pool Definition

- Storage pool updated with overflow location

```
adsm> Update stg dltpool ovflocation="RM321, BDG81"
```

- Full media volumes in library moved to overflow location

```
adsm> Move media * stg=dltpool full=y
```

  - Volumes are ejected from library and status and location updated

- Query all media volumes in storage pool

```
adsm> Query media * stg=dltpool full=n

Volume Name   State                    Location          Automated
                                                         LibName
------------- ----------------------   ---------------   --------
VOL001        Mountable in Library                       LIB01
VOL002        Mountable not in Lib.    RM321 BDG81
VOL003        Mountable not in Lib.    RM321 BDG81
VOL004        Mountable in Library                       LIB01
```

8.2.6 Sample Overflow Storage Pool Definition

This example shows the process of defining the overflow storage pool location, moving the media to that location, verifying the location of the media, and issuing a mount request to recall one of the media volumes.

8.2.6.1 Storage Pool Updated with Overflow Location

Assuming that storage pool DLTPOOL already exists, the UPDATE STGPOOL command is used to provide the pool with an overflow location. In this case the overflow location is entered within quotes because it contains embedded blanks.

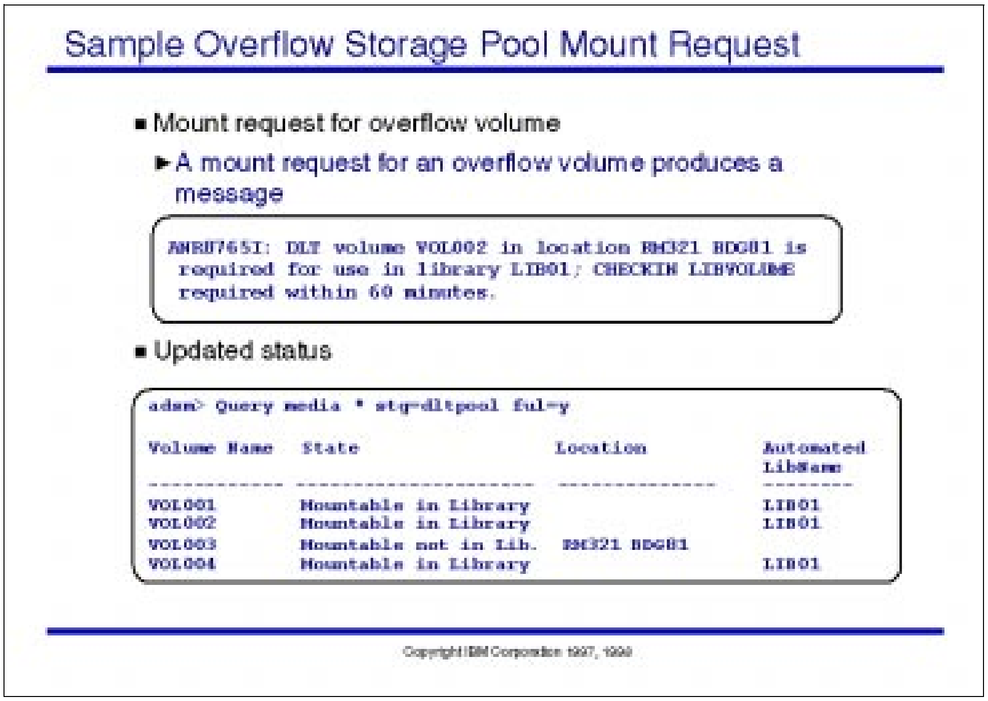
8.2.6.2 Full Media Volumes in Library Moved to Overflow Location

The MOVE MEDIA command is then issued to move all full volumes in DLTPOOL to the default overflow location for that storage pool. When this command is issued, the volumes are checked out of library LIB01, and their location and state are updated to RM321 AREA51 and *mountablenotinlib*, respectively.

8.2.6.3 Query All Media Volumes in Storage Pool

To verify that the ADSM database has been updated with the new state and location of the volumes, the QUERY MEDIA command can be issued specifying that all volumes in DLTPOOL should be shown.

The command reports that, of the four volumes in DLTPOOL, two have been moved to the overflow location and have had their states and locations updated within the ADSM database. The other two volumes in the storage pool are still in the library and ready for ADSM operations.



Sample Overflow Storage Pool Mount Request

- Mount request for overflow volume
  - A mount request for an overflow volume produces a message

```
ANR8765I: DLT volume VOL002 in location RM321 BDG01 is
 required for use in library LIB01; CHECKIN LIBVOLUME
 required within 60 minutes.
```

- Updated status

```
adsm> Query media * stg=dltpool ful=y

Volume Name   State                    Location         Automated
                                                        LibName
------------  ----------------------   --------------   --------
VOL001        Mountable in Library                      LIB01
VOL002        Mountable in Library                      LIB01
VOL003        Mountable not in Lib.     RM321 BDG01
VOL004        Mountable in Library                      LIB01
```

Copyright IBM Corporation 1997, 1998

8.2.7 Sample Overflow Storage Pool Mount Request

When storage pool volumes are in the *mountablenotinlib* state, they are still considered available for ADSM processing, and, if the volumes are needed for an operation, ADSM still generates mount requests for those volumes.

8.2.7.1 Mount Request for Overflow Volume

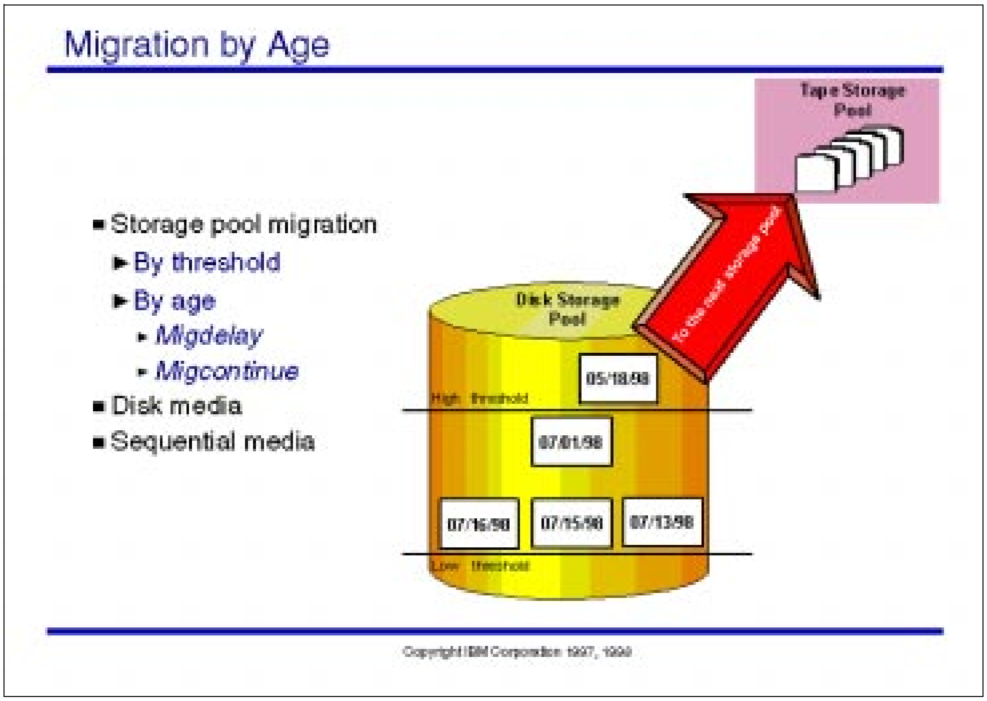
When a mount request is generated for a volume that has been moved to the overflow storage pool location, ADSM is aware that the volume is not in the library and produces a message that specifies:

- The name of the volume required
- The overflow storage pool location
- The library into which the volume is to be checked

The message states that the volume should be returned to the library and checked in through the CHECKIN LIBVOLUME command. Once the volume has been checked in, the requesting operation continues.

8.2.7.2 Updated Status

Running the QUERY MEDIA command after the volume has been returned to the library shows that both the location and status have been updated, and the volume is now ready for use.



Migration by Age

- Storage pool migration
  - By threshold
  - By age
    - *Migdelay*
    - *Migcontinue*
- Disk media
- Sequential media

Copyright IBM Corporation 1997, 1998

8.2.8 Migration by Age

ADSM Version 3 allows system administrators to specify a minimum number of days data should remain within a storage pool before it is migrated to the next storage pool (if one has been defined).

8.2.8.1 Storage Pool Migration

In ADSM storage pools are used as a repository for data backed up, archived, or migrated by a client. When defining a storage pool, it is possible to specify a next storage pool where data has to be moved when the storage pool has filled up. This is the basic concept of creating an ADSM storage pool hierarchy.

In ADSM Version 2 the storage pool migration process was based on thresholds. When defining a storage pool it was possible to specify a high migration threshold and a low migration threshold. As soon as the high migration threshold was reached, the storage pool migration process started. The migration process moved data according to the file space size of any client, that is, it moved data stored on the file spaces belonging to the client with the largest single file space in the storage pool. The migration process continued until the low threshold was reached. This migration process ensured clients continuous accessibility to the storage pool, but it gave no consideration to whether the data needed to remain in the storage pool for a specified amount of time.

In ADSM Version 3 the same migration process applies, but a new factor influences the choice of data to be migrated: the age of files residing in the storage pool. When defining a storage pool in ADSM Version 3, it is possible to specify the minimum number of days data has to remain in the storage pool before it is migrated. However, the high migration threshold is still considered first, as it was in ADSM Version 2. Once the high migration threshold has been exceeded, a check is made to see how long the data has been in the storage pool. If data has been in the storage pool for fewer days than required by the migration age value, it is not migrated. All other data exceeding the value specified is migrated as in ADSM Version 2.

Two new parameters have been added to the DEFINE STGPOOL and the UPDATE STGPOOL commands to allow system administrators to specify the minimum number of days data has to remain in a storage pool before it is migrated:

- MIGDELAY

  Specifies the number of days migration has to be delayed for files in the storage pool. This ensures files remain in the storage pool for a minimum number of days. The default for this parameter is 0, which indicates that migration is not to be delayed for files residing in this storage pool.

- MIGCONTINUE

  Specifies whether ADSM can continue the migration process for files that have not exceeded the *migdelay* value and reach the low migration threshold. The default for this parameter is YES.

8.2.8.2 Disk Media

For random access storage pools, the migration process begins when the high migration threshold has been reached, as in ADSM Version 2. First the file space using the most space in the storage pool is selected to be migrated. Only those files inserted in the storage pool by a number of days greater than the value specified by the *migdelay* parameter specified for this storage pool are selected to be migrated to the next storage pool. This process applies to all file spaces within the storage pool.

If the low migration threshold is not reached after all the eligible files have been migrated, the *migcontinue* parameter is checked. If the value for *migcontinue* has been set to YES, ADSM continues the migration process based on how long the files have been in the storage pool. The oldest files in the storage pool are migrated until the low migration threshold has been exceeded. If the value for *migcontinue* has been set to NO, the migration process ends, and a warning message is issued to notify administrators that an out-of-space condition may occur.

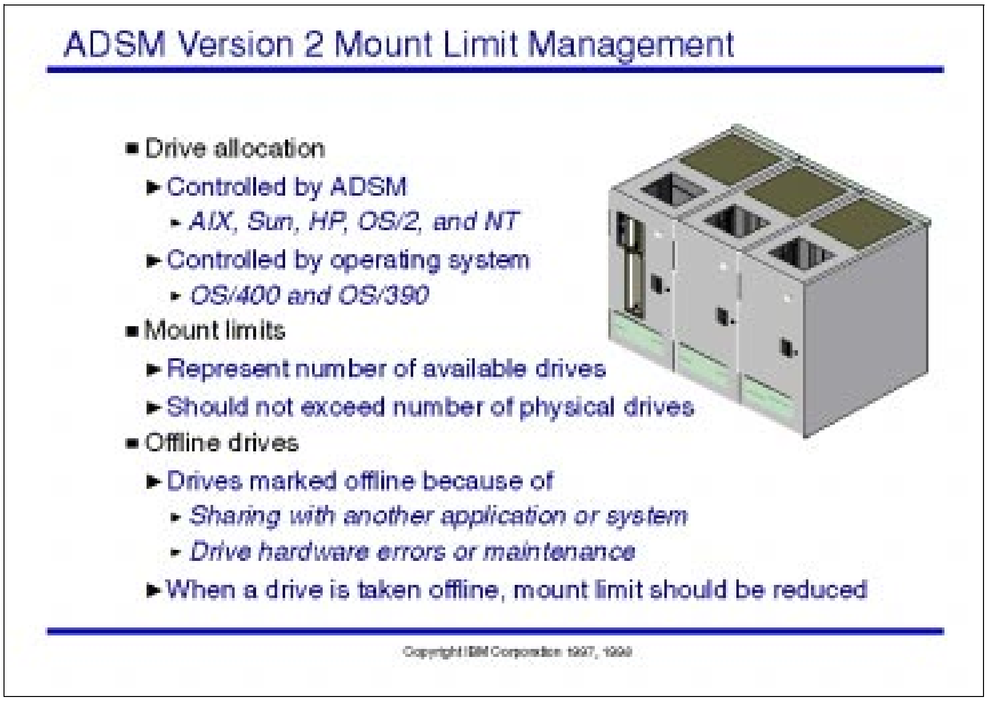
8.2.8.3 Sequential Media

For sequential storage pools, the migration process begins when the high migration threshold has been reached, as in ADSM Version 2. First, volumes that have reached the reclaim threshold specified for this storage pool are reclaimed. If after the reclamation process the low migration threshold has not been reached, the least recently referenced volume is selected as a candidate for migration. If the number of days since the volume was last written to is greater than the value specified by the *migdelay* parameter specified for this storage pool, the volume is selected for migration. All data on the volume is then migrated to the next storage pool. This process continues until the low migration threshold is reached.

Whether the low migration threshold is not reached after the last volume has been migrated, the *migcontinue* parameter is checked. If the value for *migcontinue* has been set to YES, ADSM continues the migration process by selecting volumes on the basis of least-recently-referenced criteria until the low migration threshold has been exceeded. If the value for *migcontinue* has been set to NO, the migration process ends, and a warning message is issued to notify administrators that an out-of-space condition may occur.

Copy Storage Pools

Copy storage pools are not affected by this new migration process.



ADSM Version 2 Mount Limit Management

- Drive allocation
  - Controlled by ADSM
    - *AIX, Sun, HP, OS/2, and NT*
  - Controlled by operating system
    - *OS/400 and OS/390*
- Mount limits
  - Represent number of available drives
  - Should not exceed number of physical drives
- Offline drives
  - Drives marked offline because of
    - *Sharing with another application or system*
    - *Drive hardware errors or maintenance*
  - When a drive is taken offline, mount limit should be reduced

8.2.9 ADSM Version 2 Mount Limit Management

Some new options have been introduced for handling removable storage drives within ADSM. In ADSM Version 2 the number of available drives, or mount limit, was fixed from the device class definition and had to be manually updated.

8.2.9.1 Drive Allocation

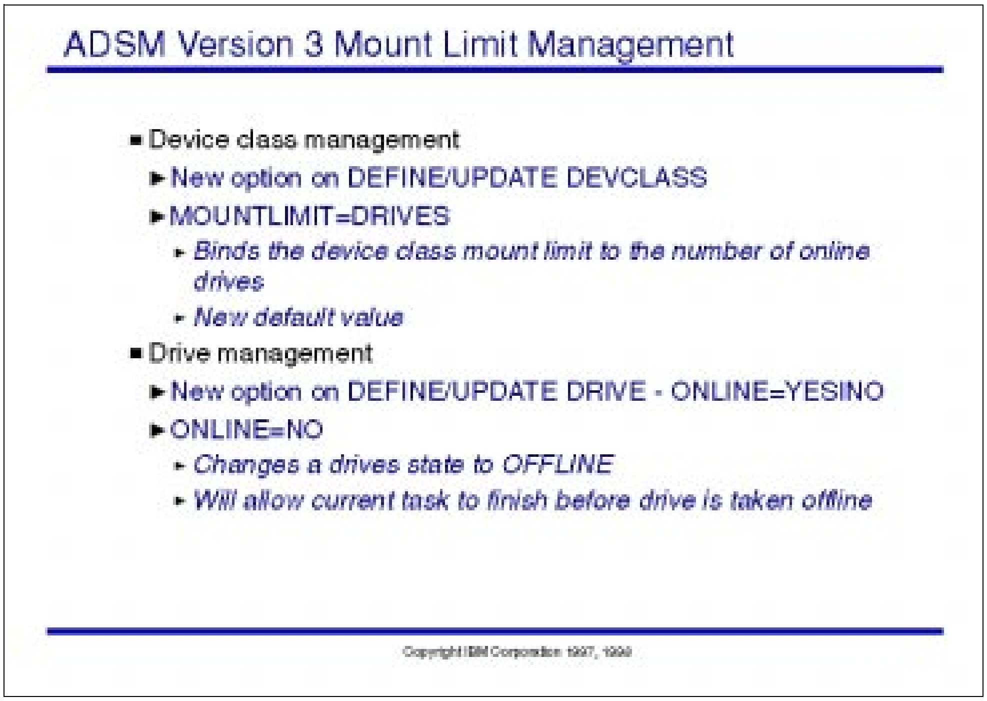
Because of the way different operating systems share their resources, drive allocation in ADSM is handled by either ADSM itself or the operating system. If ADSM is in control of drive allocation, the principal used is one of mount limits.

8.2.9.2 Mount Limits

Mount limits within ADSM are defined at the device class level. The mount limit for a particular device class represents the number of drives available for use. To prevent situations where one storage pool’s activities can interfere with another’s, it is recommended that the total number of mount limits allocated to device classes not exceed the total number of drives available for exclusive ADSM use.

8.2.9.3 Off-line Drives

There are a number of reasons why a drive may be marked off-line in ADSM and cause interference with operations. In certain libraries, drives can be shared with other applications, so that when ADSM tries to access a drive that is being used by another application, it is unable to and consequently marks the drive as being off-line. Hardware errors on a drive can also place it in the off-line state. Periodically maintenance procedures such as cleaning are required on a drive used with ADSM. To ensure that the off-line drives do not cause problems, it is necessary to adjust the mount limit accordingly.



ADSM Version 3 Mount Limit Management

- Device class management
  - New option on DEFINE/UPDATE DEVCLASS
  - MOUNTLIMIT=DRIVES
    - *Binds the device class mount limit to the number of online drives*
    - *New default value*
- Drive management
  - New option on DEFINE/UPDATE DRIVE - ONLINE=YES|NO
  - ONLINE=NO
    - *Changes a drives state to OFFLINE*
    - *Will allow current task to finish before drive is taken offline*

Copyright IBM Corporation 1997, 1998

8.2.10 ADSM Version 3 Mount Limit Management

In ADSM Version 3, a new function is introduced in the form of self-adjusting mount limits. This function allows the server to monitor the state of drives and dynamically adjust the mount limit accordingly.

8.2.10.1 Device Class Management

In ADSM Version 3, a new value has been added to the *mountlimit* parameter of the DEFINE DEVCLASS and UPDATE DEVCLASS commands. It is now possible to set *mountlimit=drives*.

When the *mountlimit* is set to *drives*, ADSM monitors the number of on-line drives within the library and adjusts the mount limit accordingly. For ADSM Version 3 this is the new default value.

8.2.10.2 Drive Management

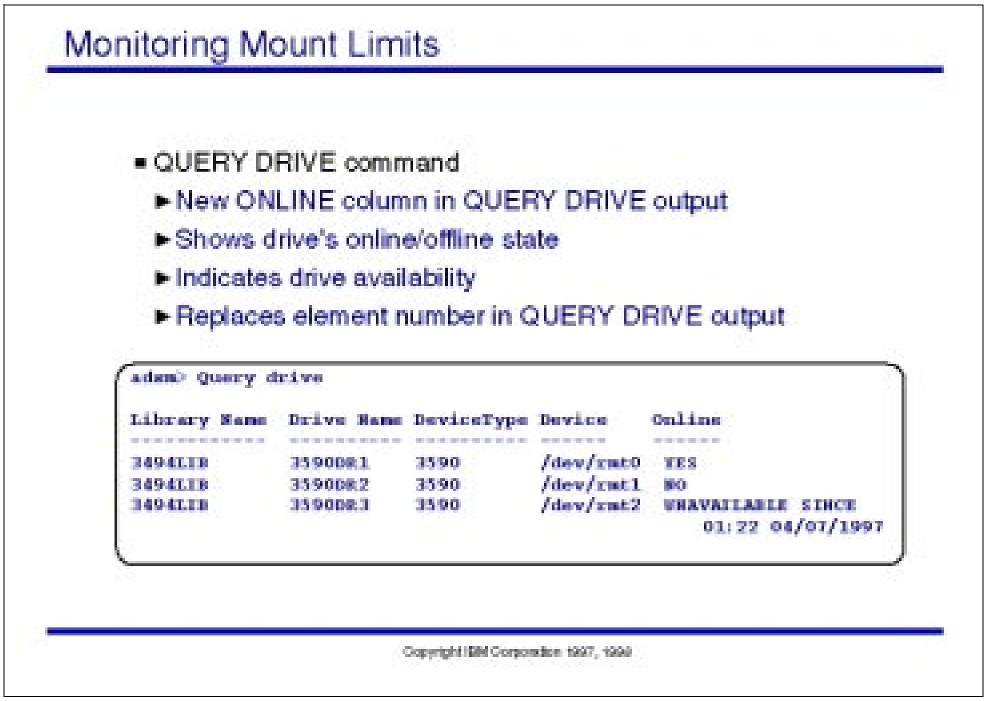
A new parameter has also been added to the DEFINE DRIVE and UPDATE DRIVE commands, it is *online=yes/no*. This parameter has been introduced to help ADSM administrators in the management of sequential storage drives.

During normal operations, each drive in the ADSM system is in an *online=yes* state. Some of the reasons why a drive may need to be removed from ADSM for a period of time include:

- Routine maintenance and cleaning
- Suspected pending hardware failure

In these cases the UPDATE DRIVE command can be issued with the parameter *online=no*. If the drive is not in use at the time the command is issued, it is placed in an off-line state. The mount limit is automatically adjusted if *mountlimit=drives* and ADSM continues without using that drive. If the drive is in use by an ADSM process when the command is issued, the tape operation continues until the process completes. When the process completes, the tape is dismounted, and the drive is immediately taken off-line.

This function allows drives to be drained from ADSM use, so that administrators can easily schedule off-line time for routine maintenance.



Monitoring Mount Limits

- QUERY DRIVE command
  - New ONLINE column in QUERY DRIVE output
  - Shows drive's online/offline state
  - Indicates drive availability
  - Replaces element number in QUERY DRIVE output

```
adsm> Query drive

Library Name  Drive Name  DeviceType Device     Online
------------  ----------  ---------- ------     ------
3494LIB       3590DR1     3590       /dev/rmt0  YES
3494LIB       3590DR2     3590       /dev/rmt1  NO
3494LIB       3590DR3     3590       /dev/rmt2  UNAVAILABLE SINCE
                                                    01:22 04/07/1997
```

8.2.11 Monitoring Mount Limits

Because ADSM is in control of the mount limits of device classes when the *mountlimit=drives* option is used, the QUERY DRIVE command has been updated to provide output for this new information.

To allow the administrator to track the new state of the ADSM drives, a column entitled On-line has been added to the QUERY DRIVE command output. This column replaces the *element* value, which is now only available in the QUERY DRIVE *format=detail* output. This column reports one of three states:

- YES - The drive is on-line and available for ADSM operations.
- NO - The drive is off-line and has been placed in that state by an administrator updating its status.
- UNAVAILABLE - This output also includes a date and time showing the last time ADSM was able to make contact with the drive.

The *unavailable* value enables administrators to see when the drive went into the unavailable state.

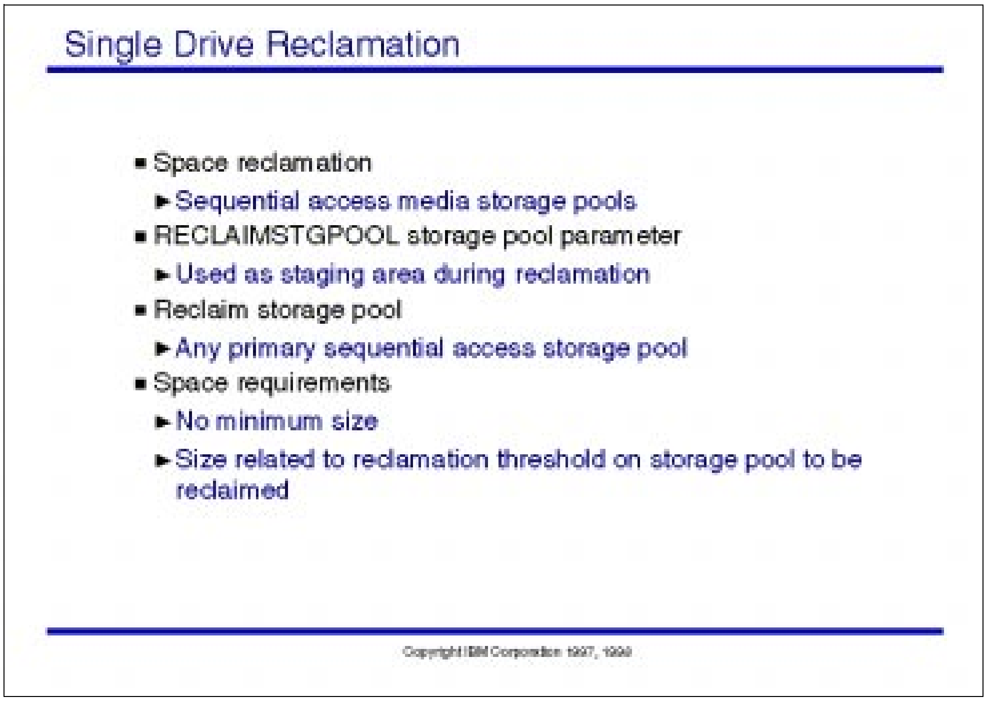
Unavailable State

The unavailable state is currently used only for drives in 3494 libraries.

In the example shown above, the QUERY DRIVE command produces output for three drives in the 3494LIB library:

- 3590DR1 - is on-line and available for server operations.

- 3590DR2 - is off-line and was made that way by an ADSM administrator issuing the UPDATE DRIVE 3494LIB 3590DR2 ONLINE=NO command.
- 3590DR3 - Has been unavailable since the time and date given. This unavailability may be due to a hardware error, routine maintenance, or use by another application.



Single Drive Reclamation

- Space reclamation
  - Sequential access media storage pools
- RECLAIMSTGPOOL storage pool parameter
  - Used as staging area during reclamation
- Reclaim storage pool
  - Any primary sequential access storage pool
- Space requirements
  - No minimum size
  - Size related to reclamation threshold on storage pool to be reclaimed

Copyright IBM Corporation 1997, 1998

8.2.12 Single Drive Reclamation

In ADSM Version 2 the process of reclamation for single drive libraries is not a function of ADSM. Single drive reclamation is possible but requires the administrator to enter a series of commands. In ADSM Version 3 single drive reclamation has been integrated into the server.

8.2.12.1 Space Reclamation

Reclamation is the process that enables ADSM to recover space freed up by the expiry of client data within sequential access media volumes. Only the active data is moved off the volume to be reclaimed and rewritten to a new volume in the storage pool. This process often leads to the data from two or more volumes being consolidated onto one volume.

8.2.12.2 RECLAIMSTGPOOL Storage Pool Parameter

A new *reclaimstgpool* parameter has been added to the DEFINE STGPOOL and UPDATE STGPOOL commands. This parameter points to another storage pool that can be used as the holding area for the data being consolidated.

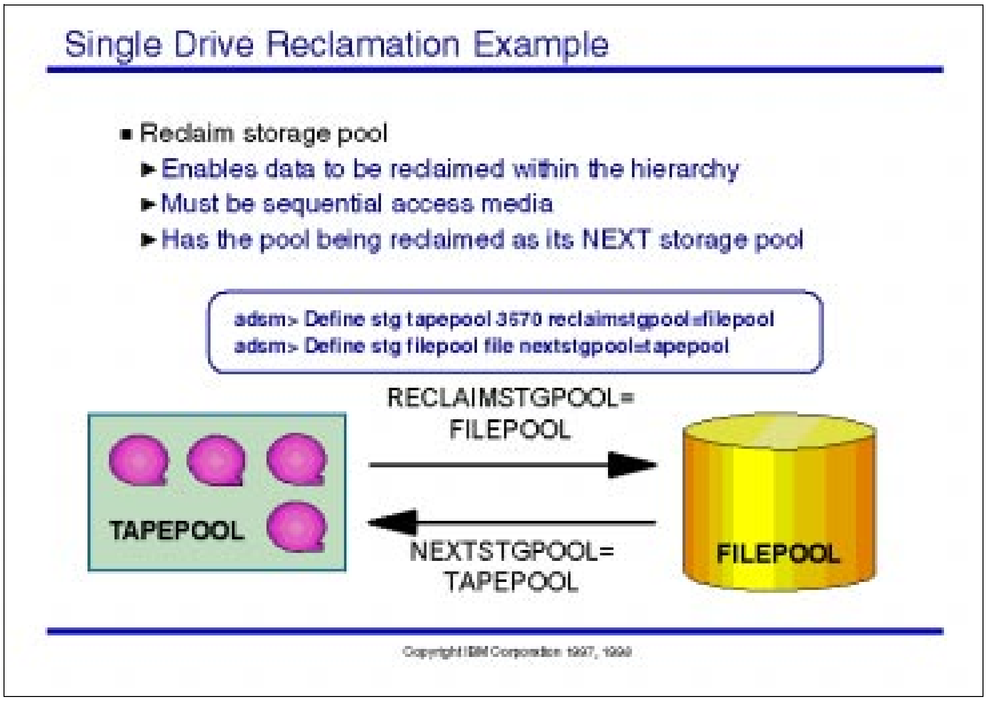
8.2.12.3 Reclaim Storage Pool

The storage pool specified as the reclaim storage pool can be any primary storage pool on the system or a new primary storage pool created for this purpose. The only disk pool allowed is one whose *devtype=file*. A copy storage pool cannot be defined as a reclaim storage pool because data can only be copied to or from a copy storage pool, not moved.

8.2.12.4 Space Requirements

When the amount of reclaimable space on a media volume exceeds the reclamation threshold, ADSM automatically begins the reclamation process. The volume to be reclaimed is mounted in the drive, and the active data is moved to the reclaim storage pool. If the reclaim storage pool is filled, the volume being reclaimed is dismounted, and a new volume in the same tape pool is mounted. The reclaimed data in the reclaim pool is then migrated to that tape volume. Once this process is complete, it repeats until all valid data has been reclaimed from the source volume being reclaimed.

If the reclaim storage pool is not filled before the source volume is emptied, another source volume is mounted, and reclamation continues. This process continues until either the reclaim storage pool is filled or all expired data within the storage pool has been removed.



Single Drive Reclamation Example

- Reclaim storage pool
  - Enables data to be reclaimed within the hierarchy
  - Must be sequential access media
  - Has the pool being reclaimed as its NEXT storage pool

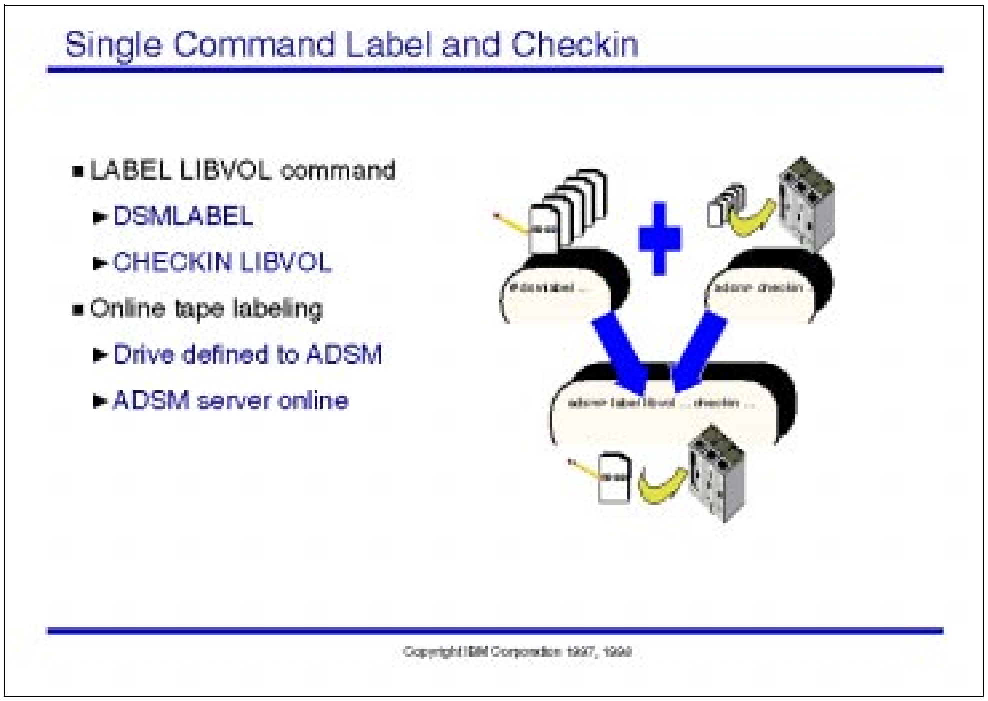
```
adsm> Define stg tapepool 3570 reclaimstgpool=filepool
adsm> Define stg filepool file nextstgpool=tapepool
```

8.2.13 Single Drive Reclamation Example

Single drive reclamation is effectively an automation of the MOVE DATA command and the storage pool migration process. The MOVE DATA process moves the remaining active data from the storage pool being reclaimed to the *reclaimstgpool*, which must be defined as sequential access media. This reclaimed data is then migrated back to a new sequential media volume in the storage pool being reclaimed.

The *reclaimstgpool* is defined with its *nextstgpool* parameter pointing back to the pool being reclaimed. Thus the reclaimed data can be migrated back to the original storage pool.

In the example, TAPEPOOL is a storage pool defined within a single drive library. TAPEPOOL is defined with FILEPOOL as its *reclaimstgpool*, and the FILEPOOL is defined with the TAPEPOOL as its *nextstgpool* storage pool.



Single Command Label and Checkin

- LABEL LIBVOL command
  - DSMLABEL
  - CHECKIN LIBVOL
- Online tape labeling
  - Drive defined to ADSM
  - ADSM server online

8.2.14 Single Command Label and Check in

ADSM Version 3 introduces the LABEL LIBVOLUME command for the purpose of labeling new media and checking them into a storage library. This command effectively combines the work of the DSMLABEL utility and the CHECKIN LIBVOLUME command.

8.2.14.1 LABEL LIBVOL Command

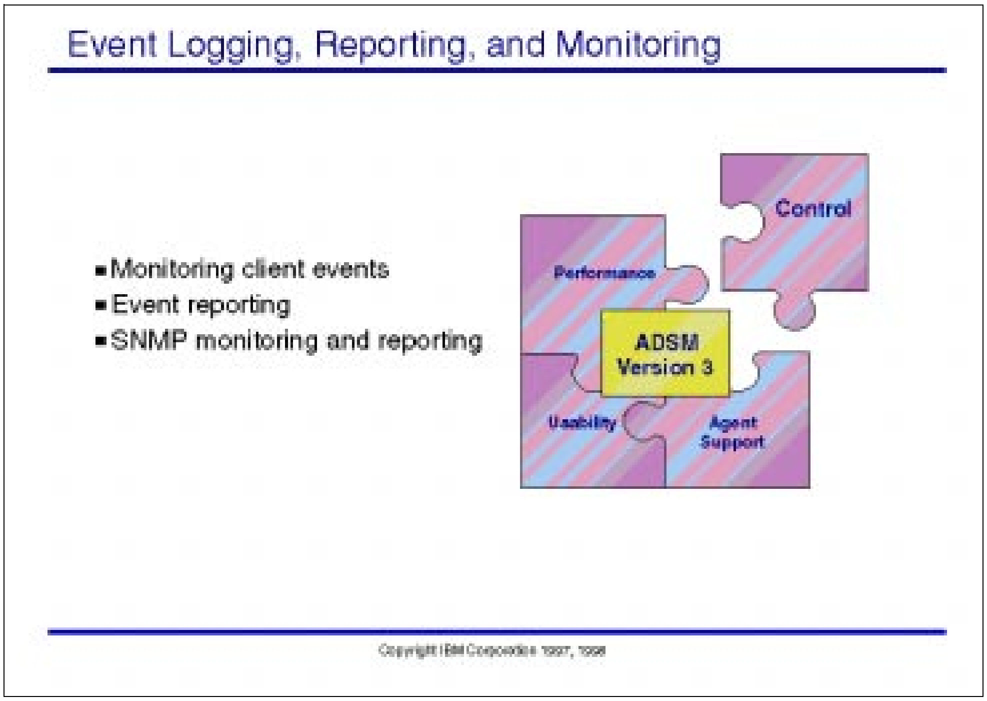
With the introduction of the new LABEL LIBVOL command, ADSM Version 3 allows the work normally requiring a DSMLABEL followed by a CHECKIN LIBVOL to be completed in one step. The time and interaction required during these two labor-intensive operations are significantly reduced. This command does not replace the previous method of DSMLABEL followed by CHECKIN LIBVOL, which can still be used.

The new command supports all of the function of DSMLABEL such as the search, barcode, and overwrite options. LABEL LIBVOL also checks the volumes into the library as either private or scratch volumes.

8.2.14.2 On-line Tape Labeling

In ADSM Version 2, to use a drive for tape labeling, it first had to be removed from ADSM, and then the stand-alone DSMLABEL utility had to be used. With the introduction of the LABEL LIBVOL command, tapes can now be labeled while the drive is on-line to ADSM. This significantly reduces the amount of interaction required to label sequential media volumes.

Chapter 9. Event Logging, Reporting, and Monitoring



Event Logging, Reporting, and Monitoring

- Monitoring client events
- Event reporting
- SNMP monitoring and reporting

Control
Performance
ADSM Version 3
Usability
Agent Support

Copyright IBM Corporation 1997, 1998

This chapter covers the new event logging and monitoring functions. It covers the following topics:

- Monitoring client events

  ADSM Version 3 provides the ability to log certain client messages as events on the ADSM server. This section explains which messages can be logged and how client event monitoring is configured.

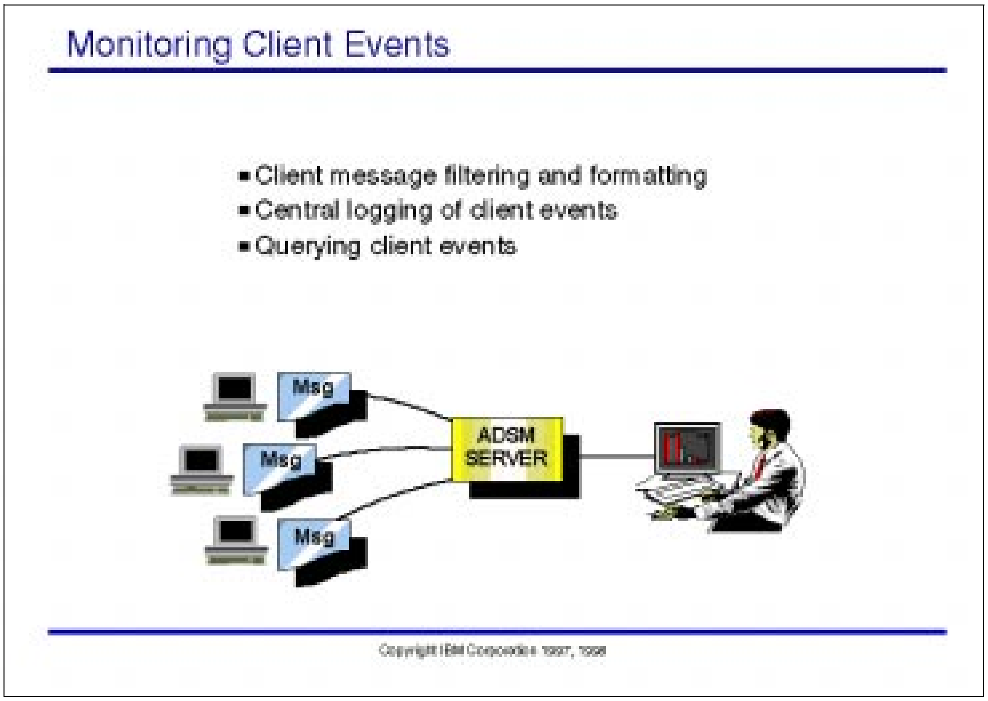
- Event Reporting

  With ADSM Version 3, client and server events can be passed to external interfaces. This section describes these interfaces and how they are configured.

- SNMP monitoring and reporting

  SNMP is a standard protocol for monitoring and reporting events within a network. ADSM Version 3 integrates ADSM into an SNMP environment. This section explains SNMP concepts and how to configure SNMP with ADSM.

9.1 Monitoring Client Events



Monitoring Client Events

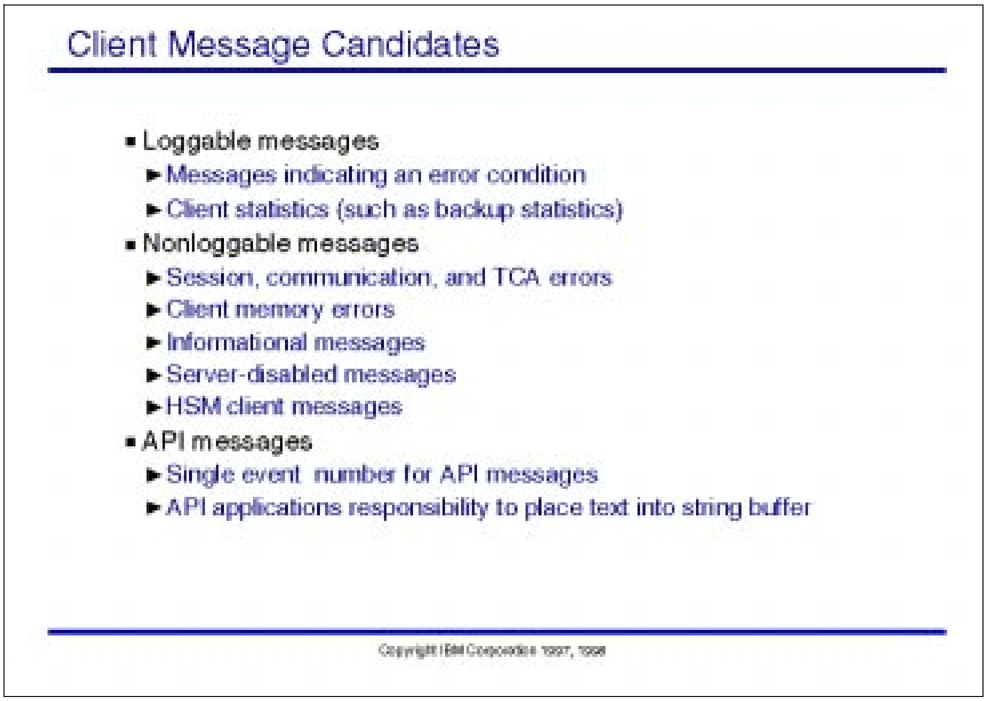
- Client message filtering and formatting
- Central logging of client events
- Querying client events

With the ADSM scheduler, client actions can be automated to run at a specific time. Messages and additional information such as backup statistics are logged locally on the client in the client schedule and error logs. To view those logs the administrator must access the client machine. One of the main benefits of central logging and monitoring is that client events can be collected and monitored at a central point.

An administrator might be interested in monitoring scheduled and nonscheduled client activity. The central logging design addresses both requirements. It includes logging of scheduled and nonscheduled client events as well as logging of server events.

This section focuses on client events and how they can be logged and monitored on the server. It covers the following topics:

- Client message filtering and formatting
- Central logging of client events
- Querying client events



Client Message Candidates

- Loggable messages
  - ►Messages indicating an error condition
  - ►Client statistics (such as backup statistics)
- Nonloggable messages
  - ►Session, communication, and TCA errors
  - ►Client memory errors
  - ►Informational messages
  - ►Server-disabled messages
  - ►HSM client messages
- API messages
  - ►Single event number for API messages
  - ►API applications responsibility to place text into string buffer

Copyright IBM Corporation 1997, 1998

9.1.1 Client Message Candidates

One of the main goals when designing the ADSM Version 3 central logging function was to provide an efficient solution. Therefore only a set of client messages are logged as events to the server.

9.1.1.1 Loggable Messages

The intention of client error logging is to notify the server of problems encountered during a client operation. Therefore client message candidates are those messages that reflect an error condition. Client statistics are also passed to the server.

9.1.1.2 Nonloggable Messages

The following message types are not logged as events:

- Session, communication, and TCA errors

  When encountering a session, communication, or a trusted communication agent (TCA) error, the client is unable to initiate a session with the server. Therefore the client cannot pass any message of this type to the server.

- Client memory errors

  Because of insufficient memory resources, the client cannot log client memory error messages.

- Informational messages

  Informational messages do not contain an error condition and therefore are not logged.

- Server disabled messages

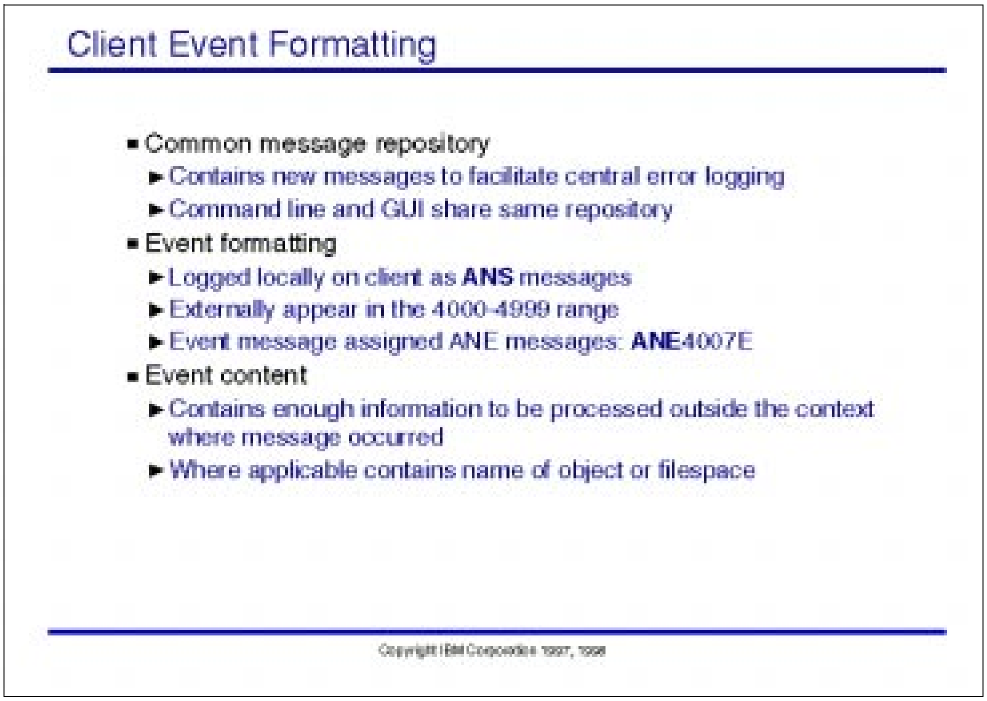
During the client signon procedure, the server provides information to the client about which messages should be logged to the server. Server-disabled messages are not passed to the server.

- HSM client messages

  Hierarchical Storage Management (HSM) client messages are not logged as events.

9.1.1.3 API Messages

For all API-related messages, a single message number is provided. The message consists of a single text string. It is the responsibility of the API application to place an appropriate message text into the string buffer.



Client Event Formatting

- Common message repository
  - Contains new messages to facilitate central error logging
  - Command line and GUI share same repository
- Event formatting
  - Logged locally on client as **ANS** messages
  - Externally appear in the 4000-4999 range
  - Event message assigned ANE messages: **ANE**4007E
- Event content
  - Contains enough information to be processed outside the context where message occurred
  - Where applicable contains name of object or filespace

9.1.2 Client Event Formatting

Client messages eligible to be sent to the server are formatted as client events to ensure that all client events have a common format.

9.1.2.1 Common Message Repository

Eligible messages are grouped in a common, shared repository. The repository resides on both the client and the server and contains new messages for all client events and related event data. The repository is shared by the command line and GUI clients. Before ADSM Version 3 the command line and GUI clients used separate repositories.

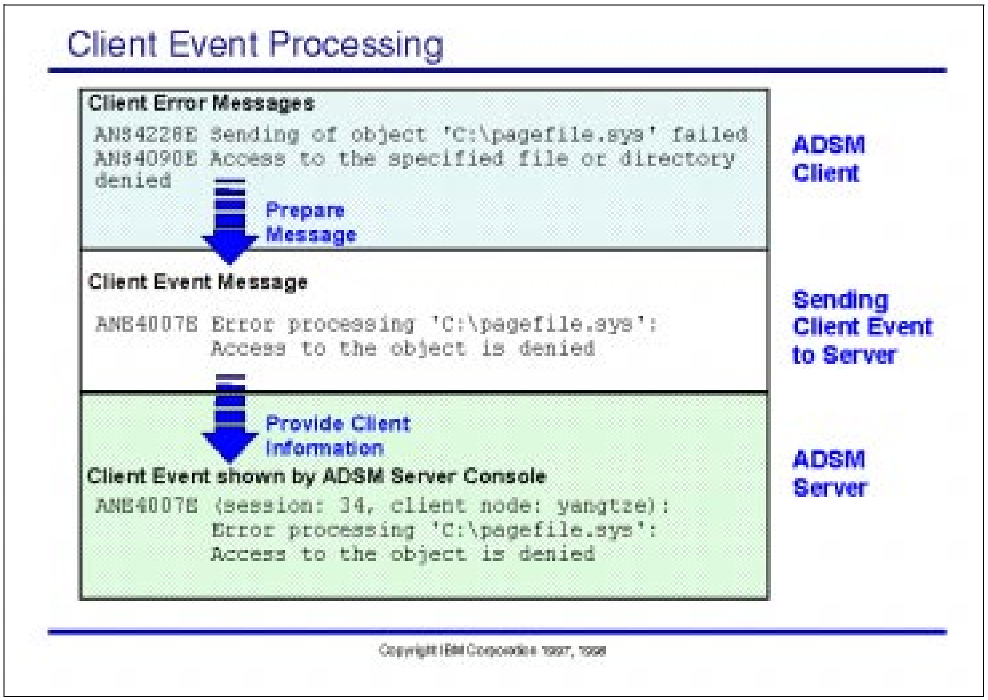
9.1.2.2 Event Formatting

Client messages in the ANS4000–ANS4999 range are eligible to be sent to the server as client events. Currently this message number range is used for command line client messages. With ADSM Version 3 these command line messages have been renumbered to ANS1000 through ANS1999 to make the 4000–4999 range available for client event logging.

Eligible client messages are sent to the server as events by using an ANE prefix instead of ANS. They appear as client events with message numbers in the ANE4000–ANE4999 range. These client messages are also logged locally in the client schedule or error logs as appropriate.

9.1.2.3 Event Content

Client event messages contain enough information to be processed outside the context where the message occurred. The client assigns the correct message number and provides information about related object or filespace names, and the server adds information such as time stamp, node name, or any other relevant information.



Client Event Processing

Client Error Messages

```
ANS4228E Sending of object 'C:\pagefile.sys' failed
ANS4090E Access to the specified file or directory
denied
```

Client Event Message

```
ANE4007E Error processing 'C:\pagefile.sys':
         Access to the object is denied
```

Client Event shown by ADSM Server Console

```
ANE4007E (session: 34, client node: yangtze):
         Error processing 'C:\pagefile.sys':
         Access to the object is denied
```

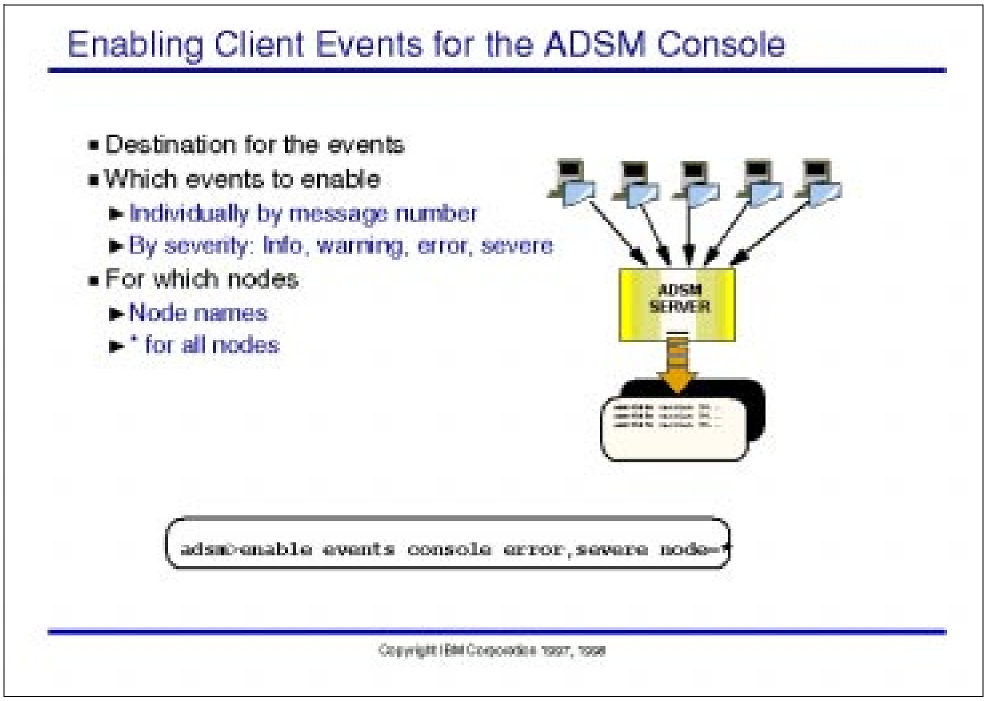
9.1.3 Client Event Processing

When an eligible client message is issued, the message number is looked up in the client message event repository and assigned the appropriate ANE message number. The event message is formatted with the related object or filespace name and is sent to the ADSM server in this format.

The server receives the event message and then adds information such as the node name from where the event was received and the session number from which the original client error message originated. If the event has been enabled for the ADSM server console, it is shown on the console as soon as all necessary information has been formatted.

National Language Support (NLS)

If different languages are being used on the clients and server, the client event messages are displayed in the language used by the ADSM server.



Enabling Client Events for the ADSM Console

- Destination for the events
- Which events to enable
  - Individually by message number
  - By severity: Info, warning, error, severe
- For which nodes
  - Node names
  - * for all nodes

```
adsm>enable events console error,severe node=*
```

Copyright IBM Corporation 1997, 1998

9.1.4 Enabling Client Events for the ADSM Console

Before client events can be passed to the server, they have to be enabled at the server. The ENABLE EVENTS command has been added to manage central logging and monitoring. It enables logging of client and server events.

This example illustrates client events that are enabled for the ADSM console. When enabling events, you have to specify the following information:

9.1.4.1 Destination for the Events

In this case the destination is the ADSM console. The console shows all related event information. For example, if the message occurred during a scheduled operation, the schedule name is also displayed.

9.1.4.2 Which Events to Enable

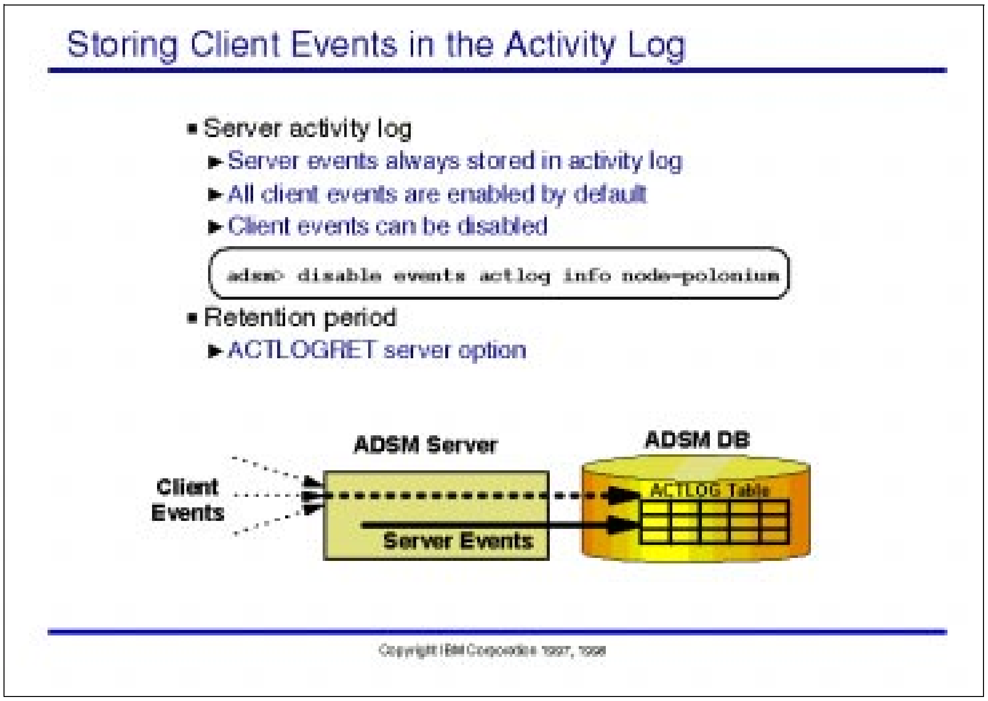
The selection can be done by a list of individual message names, severity types, or a combination of both. If all loggable messages should be tracked, specify ALL.

9.1.4.3 For Which Nodes

When client events are enabled, a node name has to be specified. You can specify only certain node names, or you can specify all, by using the wildcard character (*). To enable all nodes to log events with an ERROR or SEVERE severity to the ADSM console, use the following ENABLE EVENTS command:

```
adsm> enable events console error,severe node=*
```

Client events are displayed as soon as they have occurred on the client and have been passed to the ADSM server.



Storing Client Events in the Activity Log

- Server activity log
  - Server events always stored in activity log
  - All client events are enabled by default
  - Client events can be disabled

```
adsm> disable events actlog info node=polonium
```

- Retention period
  - ACTLOGRET server option

ADSM Server
ADSM DB
Client Events
ACTLOG Table
Server Events

Copyright IBM Corporation 1997, 1998

9.1.5 Storing Client Events in the Activity Log

The server activity log is used to store client and server events.

9.1.5.1 Server Activity Log

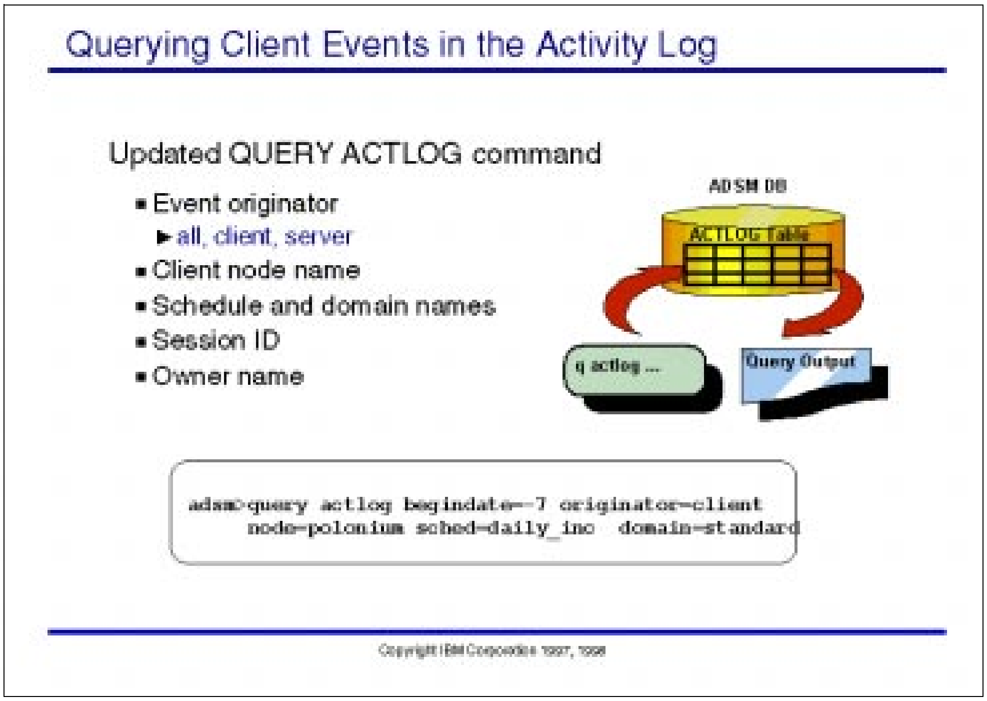
ADSM server events are always stored in the activity log and cannot be disabled because server information in the activity log is often needed to resolve critical situations. All client events are also enabled for the server activity log by default.

Client events can be disabled for the activity log. To disable information events for client node *polonium*, you would use this command:

```
adsm> disable events actlog info node=polonium
```

9.1.5.2 Retention Period

The ACTLOGRET server option specifies how long activity log records are kept in the ADSM database. Administrators should consider the ACTLOG option setting if client events are to be held in the activity log. If many client events are stored in the activity log, this will increase ADSM database usage. A short-term retention period reduces ADSM database usage, but as server and client events are deleted after the specified time frame, they cannot be queried after that. The same retention period is used for server and client events.



Querying Client Events in the Activity Log

Updated QUERY ACTLOG command

- Event originator
  - all, client, server
- Client node name
- Schedule and domain names
- Session ID
- Owner name

ADSM DB
ACTLOG Table
q actlog ...
Query Output

```
adsm>query actlog begindate=-7 originator=client
     node=polonium sched=daily_inc  domain=standard
```

Copyright IBM Corporation 1997, 1998

9.1.6 Querying Client Events in the Activity Log

The QUERY ACTLOG command has been updated to enable querying of centrally logged client events from the ADSM activity log. The following parameters have been added for extended event querying.

9.1.6.1 Event Originator

The event origin: SERVER, CLIENT, or ALL events.

9.1.6.2 Client Node Name

Client node name, to display all events that derive from this node.

9.1.6.3 Schedule and Domain Names

Schedule name and related domain name, to display all events that have been logged for the specified scheduled client activity.

9.1.6.4 Session ID

Session ID of ADSM client session, to display all logged events that originated from the specified client session.

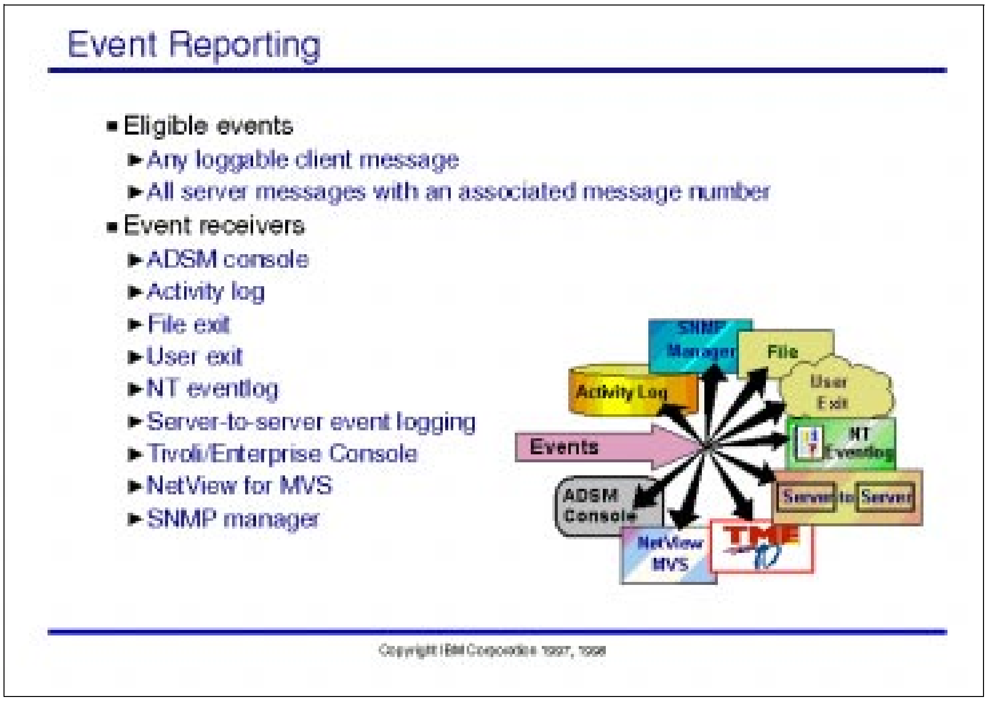
9.1.6.5 Owner Name

Owner name, to display all events that have been logged for a particular owner.

The following administrator command queries for any client events that occurred in the last seven days from node *polonium*, while running with the DAILY_INC client schedule:

```
adsm> query actlog begindate=-7 originator=client
       node=polonium sched=daily_inc
```

9.2 Event Reporting



Event Reporting

- Eligible events
  - Any loggable client message
  - All server messages with an associated message number
- Event receivers
  - ADSM console
  - Activity log
  - File exit
  - User exit
  - NT eventlog
  - Server-to-server event logging
  - Tivoli/Enterprise Console
  - NetView for MVS
  - SNMP manager

Copyright IBM Corporation 1997, 1998

This section covers the reporting of eligible client and server events to external interfaces, which are called *event receivers*.

Client messages are either loggable or nonloggable. All loggable messages can be enabled as events. There is no distinction for server events; any server message that has an associated message number is an event. All events, whether client or server, can be enabled or disabled for the event receivers. The event receivers are:

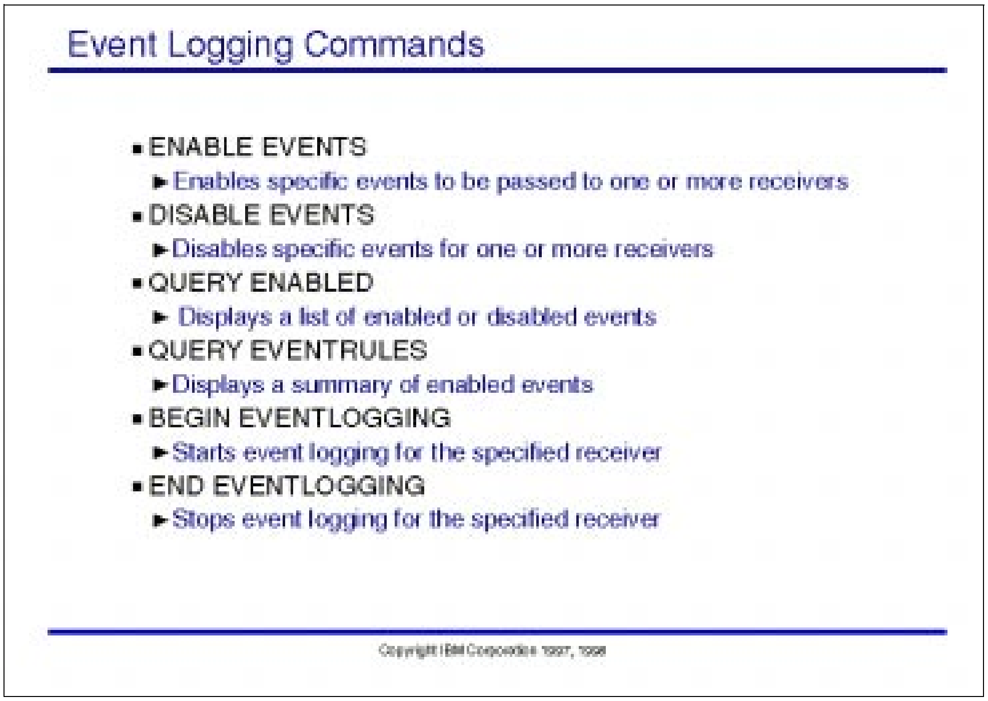
- ADSM console
- Activity log
- File exit
- User exit
- NT eventlog
- Server-to-server event logging
- Tivoli/Enterprise Console
- NetView for MVS
- SNMP manager

Events from clients or servers can be enabled to be passed to one or more receivers. It is possible to send the same events to different receivers or to enable a group of events for specific receivers. An administrator can choose any combination that suits his or her needs.

Note

All server events are enabled by default for the activity log and console receivers. They cannot be disabled for the activity log receiver. All client events are enabled by default for the activity log receiver but can be disabled if required.

The activity log and console receivers are covered in “Monitoring Client Events” on page 292. This section focuses on event receivers external to an ADSM server. It covers all external receivers with the exception of SNMP, which is covered in “SNMP Monitoring and Reporting” on page 319.



Event Logging Commands

- ENABLE EVENTS
  - ►Enables specific events to be passed to one or more receivers
- DISABLE EVENTS
  - ►Disables specific events for one or more receivers
- QUERY ENABLED
  - ► Displays a list of enabled or disabled events
- QUERY EVENTRULES
  - ►Displays a summary of enabled events
- BEGIN EVENTLOGGING
  - ►Starts event logging for the specified receiver
- END EVENTLOGGING
  - ►Stops event logging for the specified receiver

Copyright IBM Corporation 1997, 1998

9.2.1 Event Logging Commands

Some new administrative commands have been introduced to work with the event logging function.

9.2.1.1 ENABLE EVENTS

The ENABLE EVENTS command is used to enable specific events or groups of events for one or more receivers. With this command the following options are specified:

- Receiver name
- Message number or message severity
- Node name (optional)
- Server name (optional)

The following command enables all the client events for the ADSM console and the file exit receivers:

```
adsm> enable events console,file all nodename=*
```

The *nodename* parameter specifies that the client events for any node have to be sent to the receivers. To enabe individual client events, the four-digit message number must be prefixed with ANE:

```
adsm> enable event userexit ANE4027,ANE4034 node=*
```

The above command enables all nodes to send the "ANE4027 internal program message encountered" and "ANE4034 unknown system error" client events to a user exit.

To enable only server events for a receiver, omit the NODENAME parameter. Individual server messages have to be specified with the ANR prefix:

```
adsm> enable event snmp ANR0131
```

The above command enables the server error message "Server DB space exhausted" for the SNMP receiver.

The following command enables all events generated on server *adsm_severn* to be sent to the NT eventlog receiver :

```
adsm> enable event nteventlog all servername=adsm_severn
```

9.2.1.2 DISABLE EVENTS

The DISABLE EVENTS command works in the same way as the ENABLE EVENTS command, but it is used to disable, not enable, specific events. The command requires these parameters:

- Receiver name
- Message name or message severity
- Node name (optionally)
- Server name (optionally)

The parameter specifications are based on the same rules as they are for the ENABLE EVENTS command.

9.2.1.3 QUERY ENABLED

The QUERY ENABLED command displays either a list of enabled events or a list of disabled events, whichever is shorter. To display all enabled server events for the file receiver, the following command has to be issued:

```
adsm> query enabled file

All server events are disabled for the receiver file
```

In the above example, no server events are enabled for the file receiver.

To query enabled client events of node *yangtze* for the console receiver, the following command has to be issued:

```
adsm> query enabled console node=yangtze
964 client events are enabled for node YANGTZE for the CONSOLE receiver.
The following client events are DISABLED for node YANGTZE for the CONSOLE
receiver:

  ANE4000, ANE4001, ANE4002, ANE4003, ANE4004, ANE4005, ANE4006, ANE4007,
  ANE4008, ANE4009, ANE4010, ANE4011, ANE4012, ANE4013, ANE4014, ANE4015,
  ANE4016, ANE4017, ANE4018, ANE4019, ANE4020, ANE4021, ANE4022, ANE4023,
  ANE4024, ANE4025, ANE4028, ANE4029, ANE4030, ANE4031, ANE4032, ANE4033,
  ANE4034, ANE4992, ANE4993, ANE4994
```

In the above example, client events have been enabled and certain individual events have been disabled.

To query enabled server events generated on server *adsm_severn* for the NT eventlog receiver, the following command has to be issued:

```
adsm> query enabled nteventlog server=adsm_severn


All events are DISABLED for server ADSM_SEVERN for the NTEVENTLOG
receiver.
```

9.2.1.4 QUERY EVENTRULES

The QUERY EVENTRULES command displays a summary of enabled events:

```
adsm> query eventrules

Date/Time                Server Event Rules
--------------------     ------------------------------------
08/08/1997 09:21:33      ENABLE EVENTS FILE ALL
08/09/1997 13:15:29      ENABLE EVENTS SNMP ALL
08/14/1997 17:53:49      ENABLE EVENTS USEREXIT ERROR
```

Entering QUERY EVENTRULES displays the enabled server events. Entering the same command with the *nodename* parameter specified displays enabled client events for that node or nodes. Entering the same command with the *servername* parameter specified displays enabled events for the specified servers.

9.2.1.5 BEGIN EVENTLOGGING

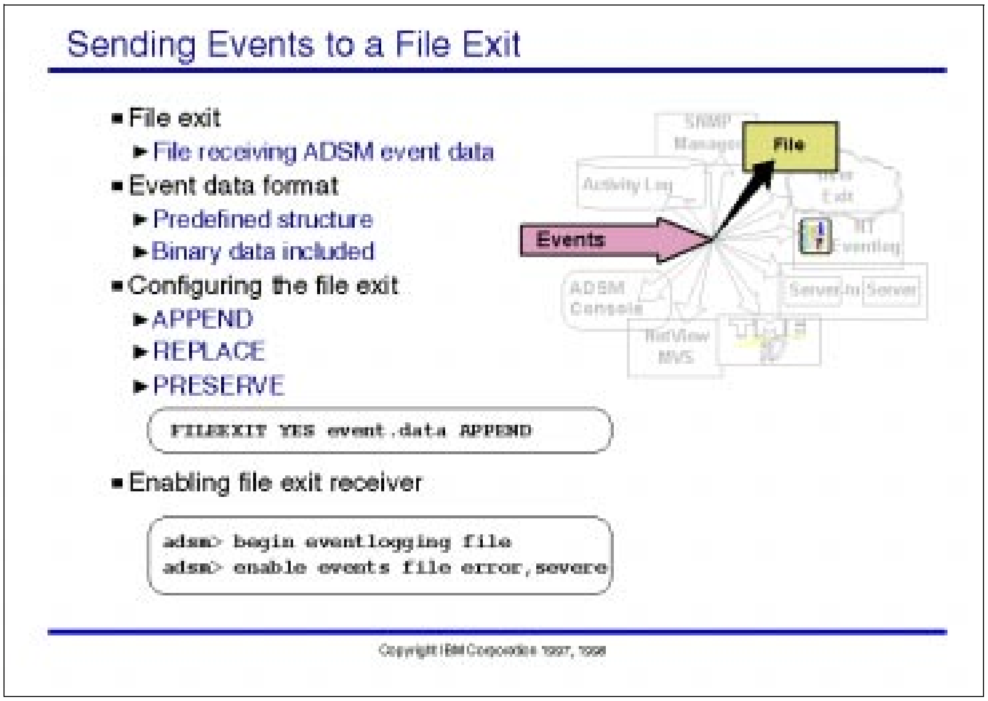
The BEGIN EVENTLOGGING command begins event logging for one or more receivers. Event logging is automatically turned on for receivers configured by default (server console and activity log). To begin event logging for the NetView receiver, enter:

```
adsm> begin eventlogging netview
```

If no receiver is specified, or if *all* is specified, logging begins for all configured receivers.

9.2.1.6 END EVENTLOGGING

The counterpart to the BEGIN EVENTLOGGING command is the END EVENTLOGGING command. It stops event logging for one or more receivers.



Sending Events to a File Exit

- File exit
  - File receiving ADSM event data
- Event data format
  - Predefined structure
  - Binary data included
- Configuring the file exit
  - APPEND
  - REPLACE
  - PRESERVE

```
FILEEXIT YES event.data APPEND
```

- Enabling file exit receiver

```
adsm> begin eventlogging file
adsm> enable events file error,severe
```

Copyright IBM Corporation 1997, 1998

9.2.2 Sending Events to a File Exit

Events can be routed to a file. The events can be client or server events depending on the administrator’s specifications for the receiver.

9.2.2.1 File Exit

A file exit is a single file capable of receiving data. ADSM writes all information related to the enabled events to this file. Because the information includes data such as event occurrence time or server name, the file can grow rapidly. This rapid growth must be considered when enabling events for a file exit.

9.2.2.2 Event Data Format

The server does not pass data in an ASCII output stream but in a data block including binary information:

```
typedef struct
{
  int32    eventNum;            /* the event number.                 */
  int16    sevCode;             /* event severity.                   */
  int16    applType;            /* application type (hsm, api, etc)  */
  int32    sessId;              /* session number                    */
  int32    version;             /* Version number of this structure (2) */
  int32    eventType;           /* event type                           */
                                 * (ADSM_CLIENT_EVENT, ADSM_SERVER_EVENT)  */
  DateTime timeStamp;           /* timestamp for event data.         */
  uchar    serverName[MAX_SERVERNAME_LENGTH+1]; /* server name       */
  uchar    nodeName[MAX_NODE_LENGTH+1]; /* Node name for session     */
  uchar    commMethod[MAX_COMMNAME_LENGTH+1]; /* communication method */
  uchar    ownerName[MAX_OWNER_LENGTH+1];     /* owner               */
  uchar    hlAddress[MAX_HL_ADDRESS+1];       /* high-level address  */
  uchar    llAddress[MAX_LL_ADDRESS+1];       /*  low-level address  */
  uchar    schedName[MAX_SCHED_LENGTH+1]; /* schedule name if applicable */
  uchar    domainName[MAX_DOMAIN_LENGTH+1]; /* domain name for node      */
  uchar    event[MAX_MSGTEXT_LENGTH];       /* event text                */
  int16    reserved1;           /* reserved field 1                      */
  int16    reserved2;           /* reserved field 2                      */
  uchar    reserved3[1400];     /* reserved field 3                      */
} eventInfo;
```

9.2.2.3 Configuring the File Exit

To activate the file exit and specify the file name, the FILEEXIT server option must be configured in the dsmserv.opt file. Use the following format:

FILEEXIT [YES **|** NO] file_name [APPEND **|** REPLACE **|** PRESERVE]

If YES is specified, event logging begins automatically at server startup. If NO is specified, event logging must be started manually with the BEGIN EVENTLOGGING command.

One of the following options can be chosen for the write operation:

APPEND If the file already exists, data will be appended to it.

REPLACE An existing file is overwritten when the exit is started.

PRESERVE If the file exists, it will not be overwritten.

In each case, the file is created if it did not exist before the file exit was activated.

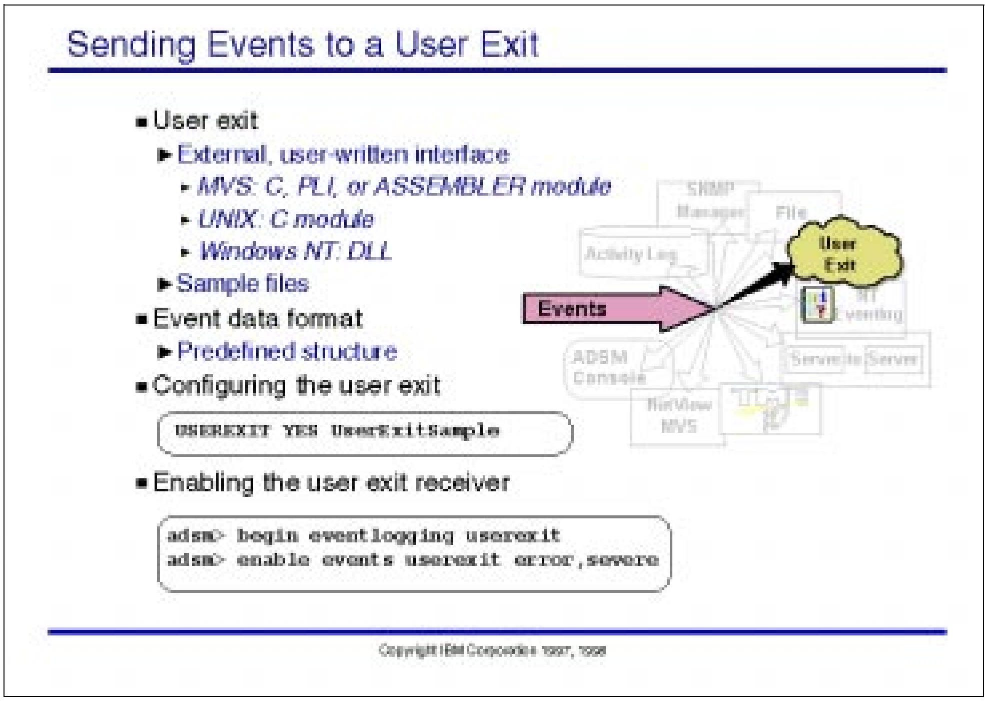
File Name

On AIX servers, if the file name specified in dsmserv.opt does not include a full path, the DSMSERV_DIR environment variable is used to determine the directory location for the file.

9.2.2.4 Enabling File Exit Receiver

Events have to be enabled for the file exit receiver before they can be passed to the specified file. Use the BEGIN EVENTLOGGING and ENABLE EVENTS commands:

```
adsm> begin eventlogging file

adsm> enable events file error,severe
```



Sending Events to a User Exit

- User exit
  - External, user-written interface
    - *MVS: C, PLI, or ASSEMBLER module*
    - *UNIX: C module*
    - *Windows NT: DLL*
  - Sample files
- Event data format
  - Predefined structure
- Configuring the user exit

```
USEREXIT YES UserExitSample
```

- Enabling the user exit receiver

```
adsm> begin eventlogging userexit
adsm> enable events userexit error,severe
```

SNMP Manager
File
Activity Log
User Exit
Events
NT Eventlog
ADSM Console
Server-to-Server
NetView MVS

Copyright IBM Corporation 1997, 1998

9.2.3 Sending Events to a User Exit

With ADSM Version 3 you can process events with a user-written program. Thus events can be processed in any form that meets customers requirements.

9.2.3.1 User Exit

A user exit is an external interface in the form of a user-written program. The program must be executable. Depending on the server platform, ADSM supports the following program types:

MVS	C, PLI, or ASSEMBLER module
UNIX	C module
Windows NT	DLL

Interpretive languages such as REXX or shell scripts are not supported.

Sample files are provided for each platform and ship with the ADSM server. For example, for the UNIX servers, four sample files are delivered: a sample C program, a sample header file, a makefile, and a reference file for compiling purposes. The C sample program does not contain any procedures but provides a complete frame and entry point for programmers to customize it.

9.2.3.2 Event Data Format

The user exit and file exit receive event data in the same data block structure (see “Sending Events to a File Exit” on page 306**)**. Besides the actual event message text, the data block contains such information as event number, host name, client name (if the message is originated on a client), the server name (if the message is originated on a different server) and originating TCP/IP address.

9.2.3.3 Configuring the User Exit

The USEREXIT server option must be configured in the dsmserv.opt file. The format differs slightly on each server platform:

MVS USEREXIT [YES **|** NO] module_name [C **|** ASSEMBLER **|** PLI]

UNIX USEREXIT [YES **|** NO] module_name

Windows NT USEREXIT [YES **|** NO] DLL_name function_name

If YES is specified, event logging begins automatically at server startup; if NO is specified, event logging must be started with the BEGIN EVENTLOGGING command. With the USEREXIT option, the name of the executable must be specified. If it is a Windows DLL, the function within the DLL must be referenced.

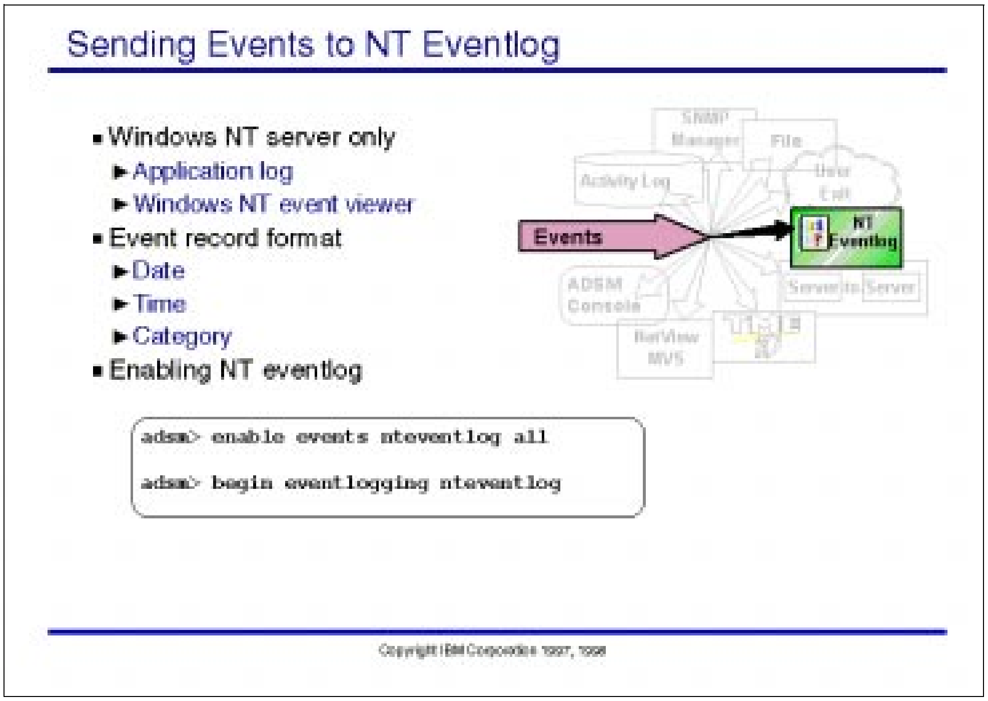
Module Name

On UNIX servers, if the module name specified in dsmserv.opt does not include a full path, the DSMSERV_DIR environment variable is used to determine the directory location for the module.

9.2.3.4 Enabling the User Exit Receiver

Events have to be enabled before they can be passed to a user exit. Use the BEGIN EVENTLOGGING and ENABLE EVENTS commands:

```
adsm> begin eventlogging userexit

adsm> enable events userexit error,severe
```



Sending Events to NT Eventlog

- Windows NT server only
  - Application log
  - Windows NT event viewer
- Event record format
  - Date
  - Time
  - Category
- Enabling NT eventlog

```
adsm> enable events nteventlog all

adsm> begin eventlogging nteventlog
```

Copyright IBM Corporation 1997, 1998

9.2.4 Sending Events to NT Eventlog

Both client and server events can be sent to the NT eventlog and viewed using the NT event viewer. Events sent to the NT eventlog depend on the definitions created for the receivers.

9.2.4.1 Windows NT Server Only

ADSM Version 3 provides the ability to choose which client and server events have to be sent to the NT eventlog. All events sent to the NT eventlog can be displayed by the NT application log through the NT event viewer.

All events with a severity class of *severe* or *error* are sent to the NT eventlog by default on any ADSM server on Windows NT. It is possible to modify this setting to add more events or to disable them.

9.2.4.2 Event Record Format

Any event in the NT eventlog is displayed by the NT event viewer as a row. A row contains the following fields:

- Date - The date as shown within the activity log
- Time - The time as shown within the activity log
- Source - Is always ADSMServer for server events. Other events, such as *ADSM client scheduler started* or *ADSM client service startup parameters*, are recorded with a source of AdsmClientService.
- Category - Four numeric digits within the ADSM message corresponding to the event. This number is associated with an ANR prefix for server events and an ANE prefix for client events.
- Event - The event number to identify the specific event

- User - Usually N/A except for *AdsmClientService* events where the user is the *NT AUTHORITY\SYSTEM*
- Computer - The name of the computer where the logged event occurred, this is the name of the computer where the ADSM server is running.

9.2.4.3 Enabling NT Eventlog

To enable or disable the events to be sent to the NT eventlog, the keyword *nteventlog* has to be used with the ENABLE EVENTS and the DISABLE EVENTS commands.

The same keyword can be specified with the BEGIN EVENTLOGGING and the END EVENTLOGGING commands to start and end event logging to the NT eventlog.

The *nteventlog* keyword can also be used with the QUERY ENABLED and QUERY EVENTRULES commands to display what is logged to the NT eventlog.

To enable more events than the default specifies, to be sent to the NT eventlog, the following commands have to be issued:

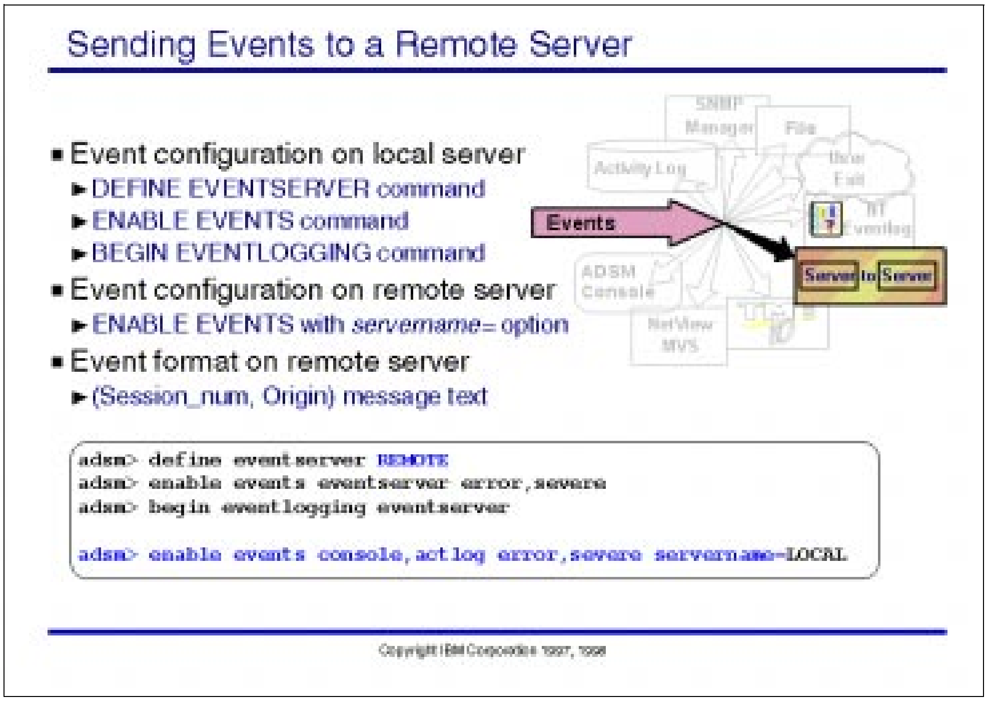
```
adsm> Enable events nteventlog all servername=*

adsm> Enable events nteventlog all nodename=*
```

The first command enables all server events generated on all known servers to be sent to the NT eventlog, and the second command enables all client events to be sent to the NT eventlog. It is possible to filter the client events by specifying the node names of the client events to be sent to the NT eventlog.

To display which events are enabled to be sent to the NT eventlog, the following commands have to be issued:

```
adsm> Query enabled nteventlog

All server events are ENABLED for the NTEVENTLOG receiver.

adsm> Query enabled nteventlog nodename=tungsten_node1

The following client events are ENABLED for node TUNGSTEN_NODE1 for the
NTEVENTLOG receiver:

  ANE4000, ANE4001, ANE4002, ANE4003, ANE4004, ANE4005, ANE4006, ANE4007,
  ANE4008, ANE4009, ANE4010, ANE4011, ANE4012, ANE4013, ANE4014, ANE4015,
  ANE4016, ANE4017, ANE4018, ANE4019, ANE4020, ANE4021, ANE4022, ANE4023,
  ANE4024, ANE4025, ANE4027, ANE4028, ANE4029, ANE4030, ANE4031, ANE4032,
  ANE4033, ANE4034, ANE4040
```

It is possible to specify the servername parameter to query the enabled events generated on a specific server. The nodename parameter of the QUERY ENABLED command must specify a defined node name, that is, it is not possible to use a generic pattern.



SNMP Manager
File
Activity Log
User Exit
NT Eventlog
Events
Server to Server
ADSM Console
NetView MVS

Sending Events to a Remote Server

- Event configuration on local server
  - DEFINE EVENTSERVER command
  - ENABLE EVENTS command
  - BEGIN EVENTLOGGING command
- Event configuration on remote server
  - ENABLE EVENTS with *servername=* option
- Event format on remote server
  - (Session_num, Origin) message text

```
adsm> define eventserver REMOTE
adsm> enable events eventserver error,severe
adsm> begin eventlogging eventserver

adsm> enable events console,actlog error,severe servername=LOCAL
```

Copyright IBM Corporation 1997, 1998

9.2.5 Sending Events to a Remote Server

Events can be logged to another server through server-to-server communications. This requires server definitions between the servers and the Enterprise Administration license enabled on both servers.

The *eventserver receiver* is the interface on the local server that receives enabled events, packages them into a verb, and sends them to an Event Server.

The *Event Server* is a server that receives events from other servers. It receives the verb sent by the EventServer receiver and routes the event to all receivers enabled for the event type and the server that sent it. This server can be the configuration manager or any other server that has been designated as the Event Server.

9.2.5.1 Event Configuration on Local Server

A series of commands is used to define the event server and then to enable the EventServer Receiver to send events to the Event Server:

1. In the EventServer receiver, define an Event Server, using the DEFINE EVENTSERVER command. This server must have been previously defined with the DEFINE SERVER command.

```
adsm> define eventserver remote
```

2. After defining an Event Server, enable the receiver to send events to the Event Server. Use the ENABLE EVENTS command to specify the desired events for the Event Server:

```
adsm> enable events eventserver error,severe
```

3. To start sending events to the Event Server, use the BEGIN EVENTLOGGING command:

```
adsm> begin eventlogging eventserver
```

9.2.5.2 Event Configuration on Remote Server

The designated Event Server must be enabled to receive events from the EventServer Receiver on other servers. The ENABLE EVENTS command is used with the SERVER parameter to specify the server from which events are to be received:

```
adsm> enable events actlog,console severe,error servername=adsm_arsenic
```

In the above example, all ERROR and SEVERE events for server adsm_arsenic are enabled to the activity log and the console on the Event Server.

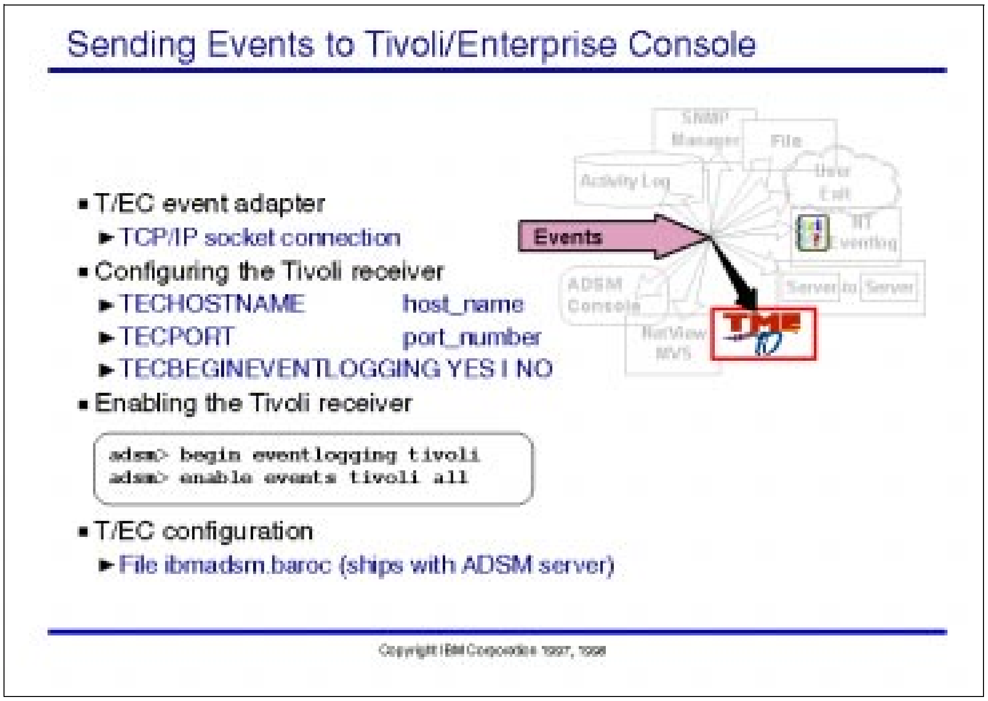
9.2.5.3 Event Format on Remote Server

Events displayed on an event server have a different format from those of the local server. These events are of the form:

(session_num, origin) message text

where *session_num* is the session number for that event; *origin*, the EventServer Receiver that sent the event; and *message text*, the description of the event.

```
ANR0407I (Session: 270, Origin: ADSM_ARSENIC)  Session 263
                         started for administrator TIMRES1 (WebBrowser) (HTTP
                         9.1.150.92(1538)).
ANR2017I (Session: 270, Origin: ADSM_ARSENIC)  Administra-
                         tor TIMRES1 issued command: AUDIT LICENSES
ANR0984I (Session: 270, Origin: ADSM_ARSENIC)  Process 11
                         for AUDIT LICENSES started in the BACKGROUND at 17:42:32.
ANR2817I (Session: 270, Origin: ADSM_ARSENIC)  AUDIT
                         LICENSES: License audit started as process 11.
ANR0609I (Session: 270, Origin: ADSM_ARSENIC)  AUDIT
                         LICENSES started as process 11.
```



Sending Events to Tivoli/Enterprise Console

- T/EC event adapter
  - TCP/IP socket connection
- Configuring the Tivoli receiver
  - TECHOSTNAME host_name
  - TECPORT port_number
  - TECBEGINEVENTLOGGING YES I NO
- Enabling the Tivoli receiver

```
adsm> begin eventlogging tivoli
adsm> enable events tivoli all
```

- T/EC configuration
  - File ibmadsm.baroc (ships with ADSM server)

Copyright IBM Corporation 1997, 1998

9.2.6 Sending Events to the Tivoli/Enterprise Console

The Tivoli/Enterprise Console (T/EC) is the TME 10 product used for monitoring and automating system activities. ADSM Version 3 provides a T/EC event adapter for sending client and server events to the T/EC. This event adapter is the Tivoli receiver.

9.2.6.1 T/EC Event Adapter

T/EC event adapters are the interfaces through which the T/EC receives events. Event adapters are typically unique to a type of event. For example, the T/EC provides event adapters for NetView/6000, HP OpenView, logfiles, and others. These event adapters can only be used to handle events from those sources. Other applications can create their own event adapters, which is what has been done with ADSM Version 3.

An event adapter has two alternate methods for sending events to the T/EC. A "secure" connection makes use of the object request broker technology within the Tivoli framework. This method requires that both the originating system and the T/EC are Tivoli-managed nodes. The second method is a direct TCP/IP socket connection. This is termed an "unsecure" connection by Tivoli as the originating system does not have to be a managed node. ADSM Version 3 uses this direct TCP/IP socket connection as it provides greater flexibility, supports all ADSM server platforms, and removes the requirement that the ADSM server must be a Tivoli-managed node.

9.2.6.2 Configuring the Tivoli Receiver

The following server options are available to define the Tivoli receiver:

TECHOSTNAME Name of host running the Tivoli Event Server

TECPORT TCP/IP port number on the Tivoli Event Server that listens for ADSM T/EC communications

TECBEGINEVENTLOGGING YES **|** NO

The TECBEGINEVENTLOGGING option specifies whether event logging for the TIVOLI receiver is started automatically during server startup or has to be started manually by the BEGIN EVENTLOGGING command.

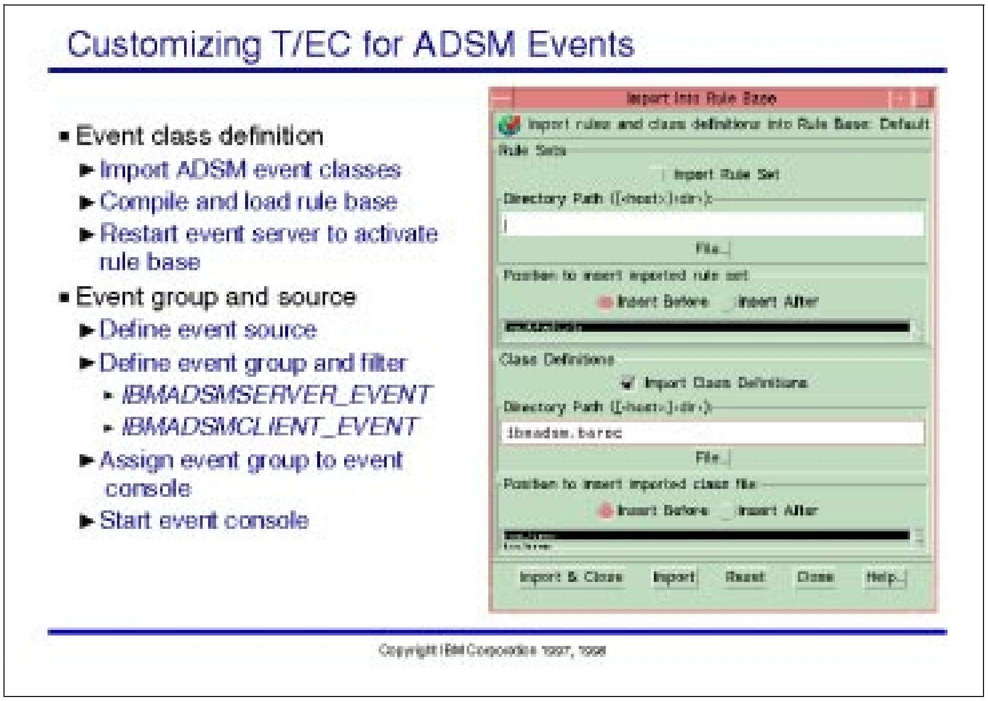
9.2.6.3 Enabling the Tivoli Receiver

The BEGIN EVENTLOGGING and ENABLE EVENTS commands must be used to specify the desired events for the Tivoli receiver:

```
adsm> begin eventlogging tivoli

adsm> enable events tivoli all
```

9.2.6.4 T/EC Configuration

The T/EC must be customized to receive and process ADSM events. Customization involves defining the ADSM event classes to the T/EC. These event classes are provided in a T/EC baroc file, IBMADSM.BAROC, provided with the ADSM server.



Customizing T/EC for ADSM Events

- Event class definition
  - Import ADSM event classes
  - Compile and load rule base
  - Restart event server to activate rule base
- Event group and source
  - Define event source
  - Define event group and filter
    - *IBMADSMSERVER_EVENT*
    - *IBMADSMCLIENT_EVENT*
  - Assign event group to event console
  - Start event console

Import Into Rule Base

Import rules and class definitions into Rule Base: Default

Rule Sets

□ Import Rule Set

Directory Path ([<host>:]<dir>):

File...

Position to insert imported rule set

● Insert Before ○ Insert After

Class Definitions

☑ Import Class Definitions

Directory Path ([<host>:]<dir>):

ibmadsm.baroc

File...

Position to insert imported class file

● Insert Before ○ Insert After

Copyright IBM Corporation 1997, 1998

9.2.7 Customizing T/EC for ADSM Events

Before ADSM events can be reported to the T/EC, you have to perform some customization steps. Import the ADSM event classes into the T/EC and configure appropriate event groups, sources, and event consoles.

9.2.7.1 Event Class Definition

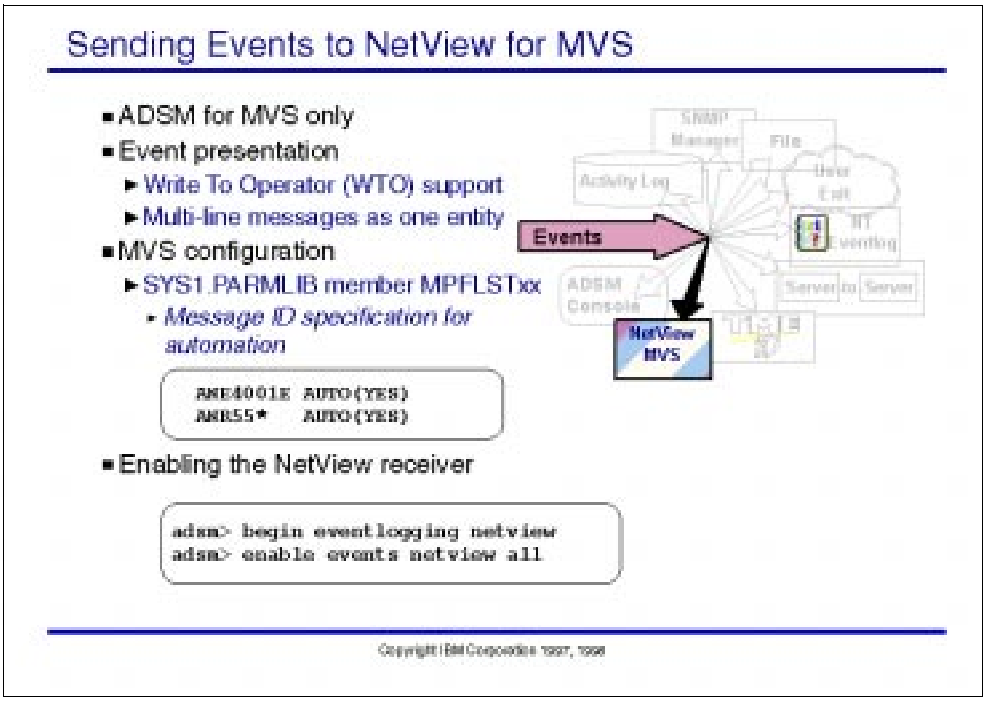
To integrate ADSM events into the T/EC, ADSM Version 3 provides T/EC event classes defined in the IBMADSM.BAROC file. These events classes have to be defined on the T/EC server by importing the IBMADSM.BAROC file into the active T/EC rule base:

1. From the TME administrator's desktop, select the current rule base by right clicking on its icon, and then select import from the menu displayed.
2. Import the IBMADSM.BAROC file into the rule base class definitions.
3. After importing the event information, the rule base must be compiled and reloaded and the T/EC server restarted.

9.2.7.2 Event Group and Source

Having imported the event definitions into the rule base, you must define a new event source and event group and configure them for use by an event console:

1. In the event server, define a new event source for ADSM.
2. After defining the event source, define a new event group and filter for ADSM. This filter must include IBMADSMSERVER_EVENT and IBMADSMCLIENT_EVENT, the two event classes previously imported from the IBMADSM.BAROC file.
3. Assign the new event group to an event console for displaying ADSM Version 3 events.
4. Start the assigned event console to display the ADSM Version 3 events.



Sending Events to NetView for MVS

- ADSM for MVS only
- Event presentation
  - Write To Operator (WTO) support
  - Multi-line messages as one entity
- MVS configuration
  - SYS1.PARMLIB member MPFLSTxx
    - *Message ID specification for automation*

```
ANE4001E AUTO(YES)
ANR55*   AUTO(YES)
```

- Enabling the NetView receiver

```
adsm> begin eventlogging netview
adsm> enable events netview all
```

SNMP Manager
File
Activity Log
User Exit
Events
TT Eventing
ADSM Console
Server to Server
NetView MVS

9.2.8 Sending Events to NetView for MVS

ADSM Version 3 provides the capability to use NetView for MVS as an event receiver. This enables central event monitoring and reporting by NetView for MVS or other compatible products.

9.2.8.1 ADSM for MVS Only

The use of NetView for MVS for event reporting is supported for the ADSM server on MVS only.

9.2.8.2 Event Presentation

An event that is forwarded to the NetView receiver is formatted for display by the MVS write to operator (WTO) support. Multiline messages are presented to NetView as one entity. WTO events are handled by the MVS message processing facility (MPF) and passed on to NetView through the MVS subsystem interface.

9.2.8.3 MVS Configuration

NetView for MVS runs in an MVS address space and receives messages from the MPF through the subsystem interface. Before messages can flow to NetView, MPF must be configured to forward them. This configuration is done by defining the message numbers in SYS1.PARMLIB member MPFLSTxx, using the AUTO(YES) parameter, which tells MPF to forward the messages onto NetView for automation:

```
ANE4001E  AUTO(YES)
ANR55*    AUTO(YES)
```

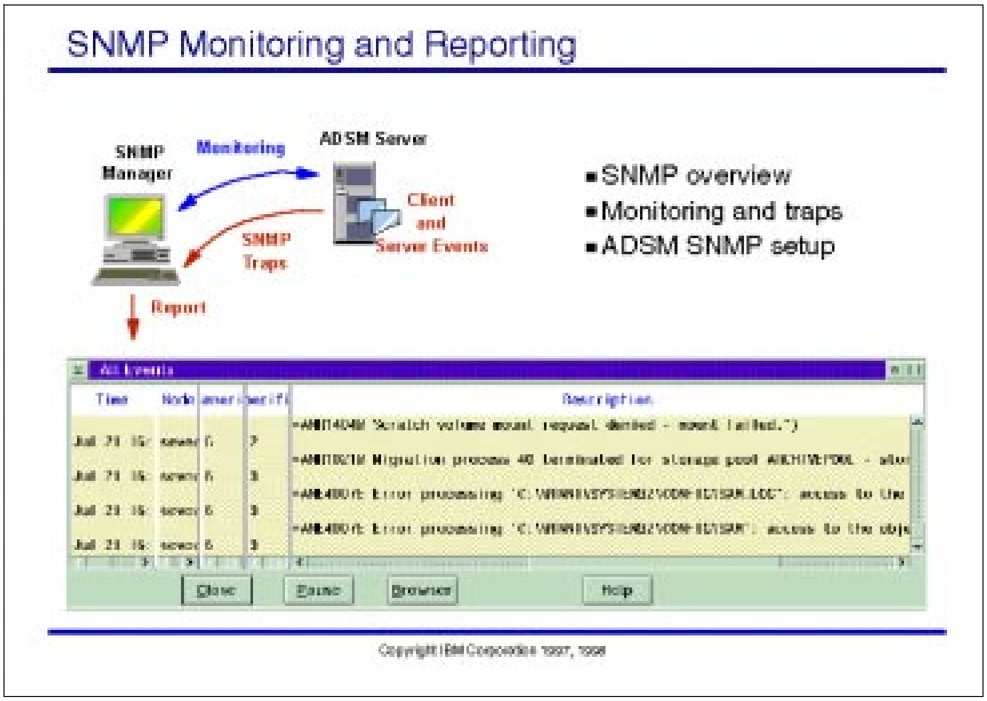
The above sample MPFLST member will forward ANE4001E client events and all server events starting with ANR55 to NetView.

9.2.8.4 Enabling the NetView Receiver

For events to be reported to NetView, each event has to be enabled for the NetView receiver by message number or by severity. Use the BEGIN EVENTLOGGING and ENABLE EVENTS commands

```
adsm> begin eventlogging netview

adsm> enable events netview all
```

9.3 SNMP Monitoring and Reporting



SNMP Monitoring and Reporting

```
Time          Node   Server  Severity  Description
                                       =ANR1404W "Scratch volume mount request denied - mount failed.")
Jul 21 16:    server 6       ?
                                       =ANR1021W Migration process 40 terminated for storage pool ARCHIVEPOOL - also
Jul 21 16:    server 6       3
                                       =ANR4[illegible] Error processing 'C:\[illegible].LOG': access to the
Jul 21 16:    server 6       3
                                       =ANR4[illegible] Error processing 'C:\[illegible]': access to the obje
Jul 21 16:    server 6       3
```

- SNMP overview
- Monitoring and traps
- ADSM SNMP setup

Copyright IBM Corporation 1997, 1998

SNMP is used to manage network elements from a central point. Network elements can either be a piece of hardware or a software application. SNMP is an industry-standard protocol and has become popular because of its simplicity of information interchange. To enable ADSM to take advantage of SNMP event monitoring and reporting, the ADSM Version 3 servers for AIX, HP-UX, Solaris, and Windows NT provide an SNMP interface. The following topics are covered:

- SNMP overview

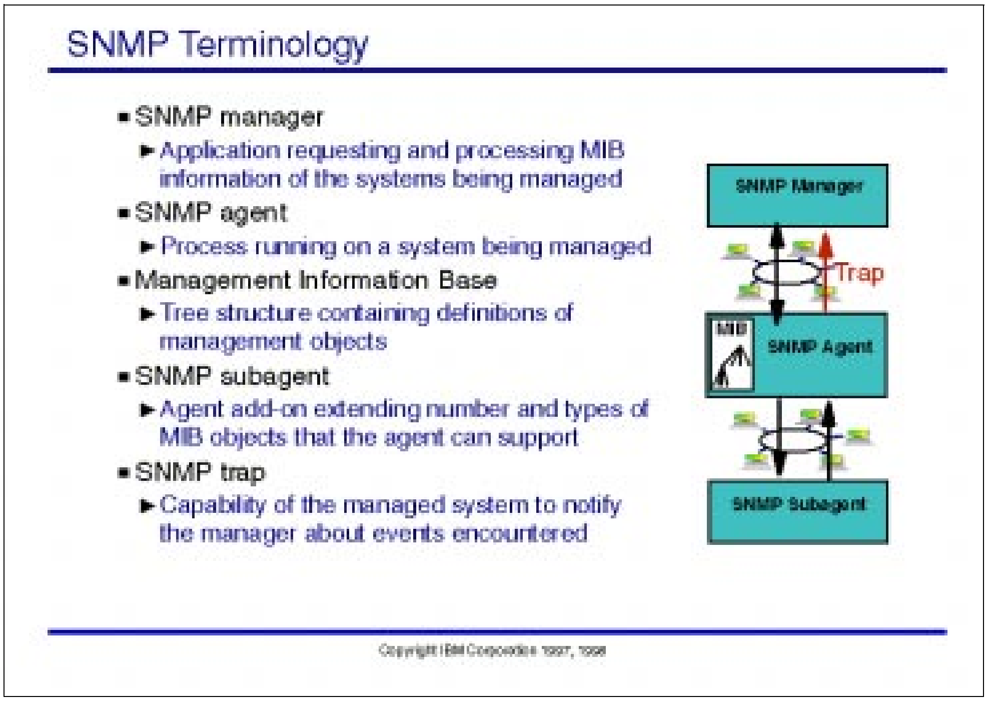
  An introduction to basic terms, components, and communication methods is provided. This section is intended for people without an SNMP background who want to become familiar with SNMP terminology and understand how ADSM can participate in an SNMP environment.

- Monitoring and traps

  Management through SNMP allows a system to be actively monitored by setting SNMP variables that correspond to management functions on the ADSM server. In addition, a heartbeat monitor is used to determine whether the ADSM server is still running. SNMP also provides the ability to send asynchronous messages, known as *traps*, to an SNMP manager. Traps are used to inform the SNMP manager about events taking place on the ADSM server. This section describes the configuration required to enable ADSM to send traps and heartbeat information to an SNMP manager.

- ADSM SNMP setup

  To configure an ADSM SNMP environment several steps must be performed on the managing system, the managed system, and the interfaces in between. This section covers those steps for a sample environment.



SNMP Terminology

- SNMP manager
  - ► Application requesting and processing MIB information of the systems being managed
- SNMP agent
  - ► Process running on a system being managed
- Management Information Base
  - ► Tree structure containing definitions of management objects
- SNMP subagent
  - ► Agent add-on extending number and types of MIB objects that the agent can support
- SNMP trap
  - ► Capability of the managed system to notify the manager about events encountered

9.3.1 SNMP Terminology

SNMP was originally designed to manage TCP/IP networks. A defined network management station monitors and controls network elements such as computers, gateways, and terminal servers. SNMP is used to communicate information between the management station and the managed network elements. The simplicity of using a very limited set of management functions made SNMP quite popular and caused its use to be extended to manage other hardware and software products.

9.3.1.1 SNMP Manager

The SNMP manager is an application that manages elements of a network or software products by sending requests to the managed systems and processing the responses. The requests can either retrieve or change information. Examples of SNMP managers are NetView/6000, NetView for OS/2, and the T/EC.

9.3.1.2 SNMP Agent

To enable communication between the SNMP manager and managed systems, each managed system runs a process known as the *SNMP agent*. The agent answers requests generated by the manager. It can also send unsolicited messages to notify the manager about an event encountered on the managed system.

9.3.1.3 Management Information Base

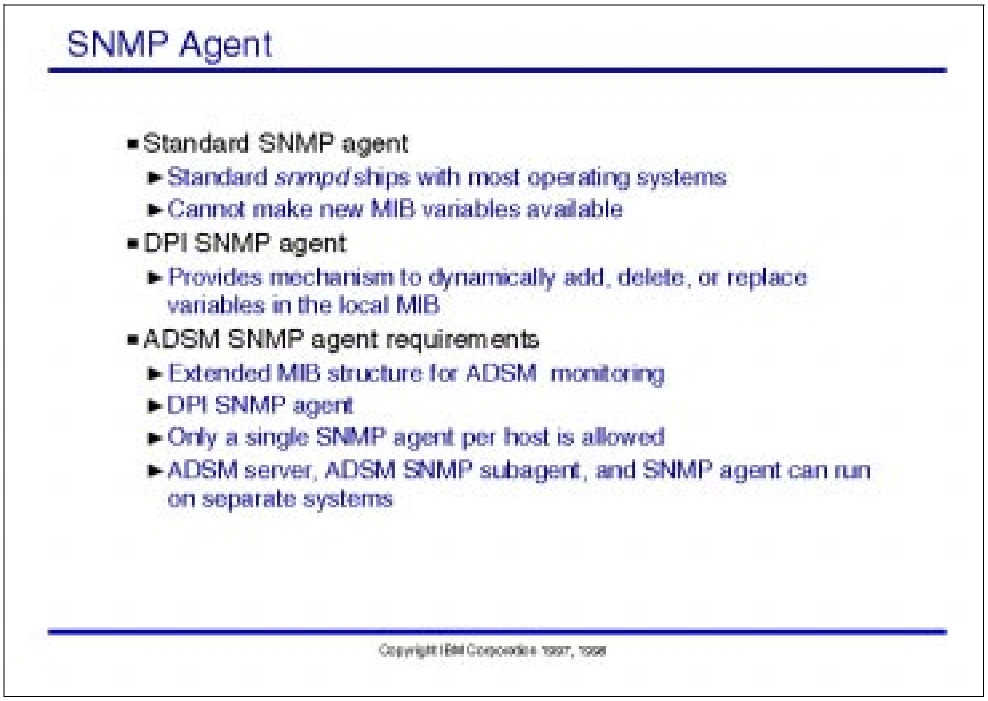
To answer manager requests, the agent looks up the requested information in its management information base (MIB). The MIB contains objects and their values in the form of a tree structure. The MIB structure is predefined and contains a limited number of objects. The structure can be extended by adding new MIB objects or subtrees.

9.3.1.4 SNMP Subagent

An SNMP subagent is used to extend the number and types of MIB objects by registering new objects or subtrees in the agent’s MIB. Such extension is necessary when SNMP is used to manage hardware or software that it was not originally designed to manage.

9.3.1.5 SNMP Trap

Traps are unsolicited, asynchronous messages, that the agent can send to the SNMP manager. Traps enable managed network elements to generate events such as agent initialization and agent restart to the SNMP manager.



SNMP Agent

- Standard SNMP agent
  - Standard *snmpd* ships with most operating systems
  - Cannot make new MIB variables available
- DPI SNMP agent
  - Provides mechanism to dynamically add, delete, or replace variables in the local MIB
- ADSM SNMP agent requirements
  - Extended MIB structure for ADSM monitoring
  - DPI SNMP agent
  - Only a single SNMP agent per host is allowed
  - ADSM server, ADSM SNMP subagent, and SNMP agent can run on separate systems

Copyright IBM Corporation 1997, 1998

9.3.2 SNMP Agent

To enable communication between an SNMP manager and its managed systems, an SNMP agent is needed. The agent, run on the managed system, answers the manager's requests and sends traps to inform the SNMP manager about events taking place on the managed system.

9.3.2.1 Standard SNMP Agent

The standard SNMP agent supplied with a system such as AIX or Windows NT is implemented through the snmpd daemon. The daemon may ship as part of the operating system or the TCP/IP product. SNMP agents can answer the standard SNMP requests and send traps to the SNMP manager. Standard agents, such as the AIX snmpd, typically have predefined MIB objects that they access. If managing a particular software requires additional MIB objects, they have to be added to the agent's MIB. Standard SNMP agents have no means of registering new objects in the MIB.

9.3.2.2 DPI SNMP Agent

The Distributed Protocol Interface (DPI) is an extension of the SNMP agent. Agents that support DPI provide a mechanism to dynamically add, delete, or replace objects in the local MIB.

9.3.2.3 ADSM SNMP Agent Requirements

ADSM management through SNMP requires additional information in the MIB of the local agent. Therefore an SNMP agent supporting DPI Version 2 must be used to communicate with the ADSM subagent.

The SNMP agent is not included with ADSM. The standard SNMP agents supplied with AIX, HP-UX, Solaris, and Windows NT do not support the DPI. Therefore, an alternative DPI SNMP agent must be installed for use with ADSM.

DPI SNMP agents ship with SystemView and Tivoli products and can also be downloaded for free from the following Web page:

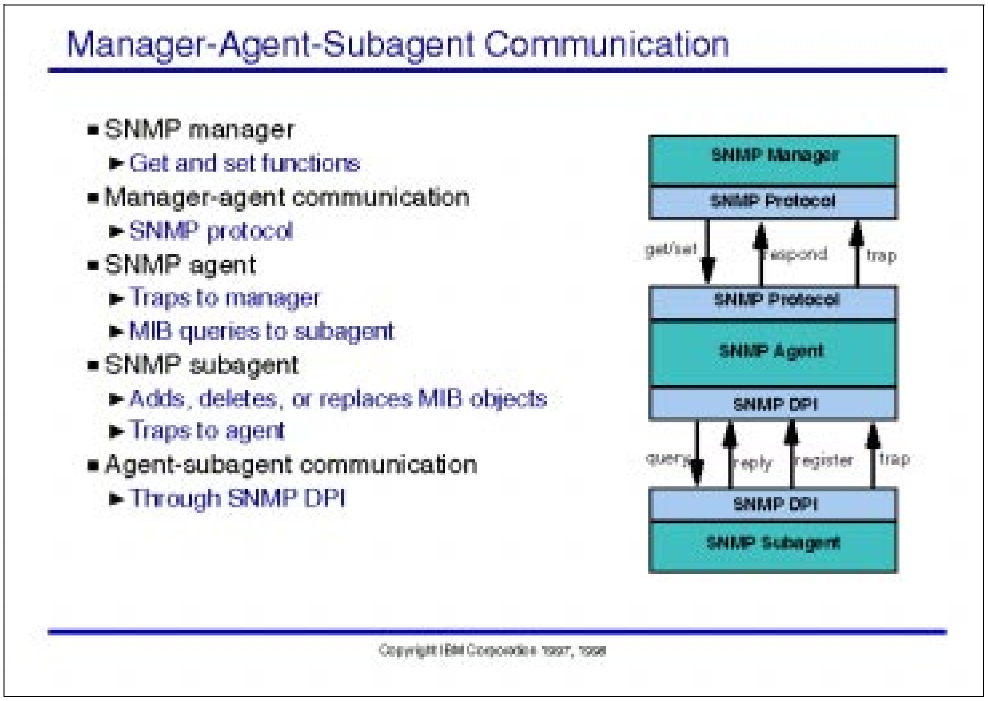
http://www.support.tivoli.com/sva/index.html

SystemView DPI SNMP agents are available at this site for AIX and Windows NT only. They are not available for HP-UX or Solaris. The ADSM server, ADSM SNMP subagent, and SNMP agent can be run on separate hosts. Therefore, for platforms without a DPI SNMP agent, the DPI SNMP agent can be run on another supported platform.

AIX Version 4.2.1

AIX Version 4.2.1 and above ships with a DPI-enabled SNMP agent. Earlier releases of AIX do not provide a DPI-enabled agent and must use the SystemView SNMP agent.

The ADSM subagent is included with ADSM and is started as a separate process communicating with the SNMP agent. Only one SNMP agent and subagent per host can be run.



Manager-Agent-Subagent Communication

- SNMP manager
  - Get and set functions
- Manager-agent communication
  - SNMP protocol
- SNMP agent
  - Traps to manager
  - MIB queries to subagent
- SNMP subagent
  - Adds, deletes, or replaces MIB objects
  - Traps to agent
- Agent-subagent communication
  - Through SNMP DPI

9.3.3 Manager-Agent-Subagent Communication

Communication between manager and agent and between agent and subagent takes place over different types of connections. Since these connections permit different types of operations, each instance of manager, agent, or subagent supports a specific set of functions.

9.3.3.1 SNMP Manager

The SNMP manager supports a limited set of get and set functions that allows it to request or change information in the agent's MIB.

9.3.3.2 Manager-Agent Communication

The SNMP manager and agent communicate with each other through the SNMP protocol. The manager passes all requests for variables to the agent, although it might be a variable of the subagent's. The agent then passes the request to the subagent and sends the answer back to the manager. The SNMP manager does not have any knowledge that the agent calls on other processes to obtain an answer.

9.3.3.3 SNMP Agent

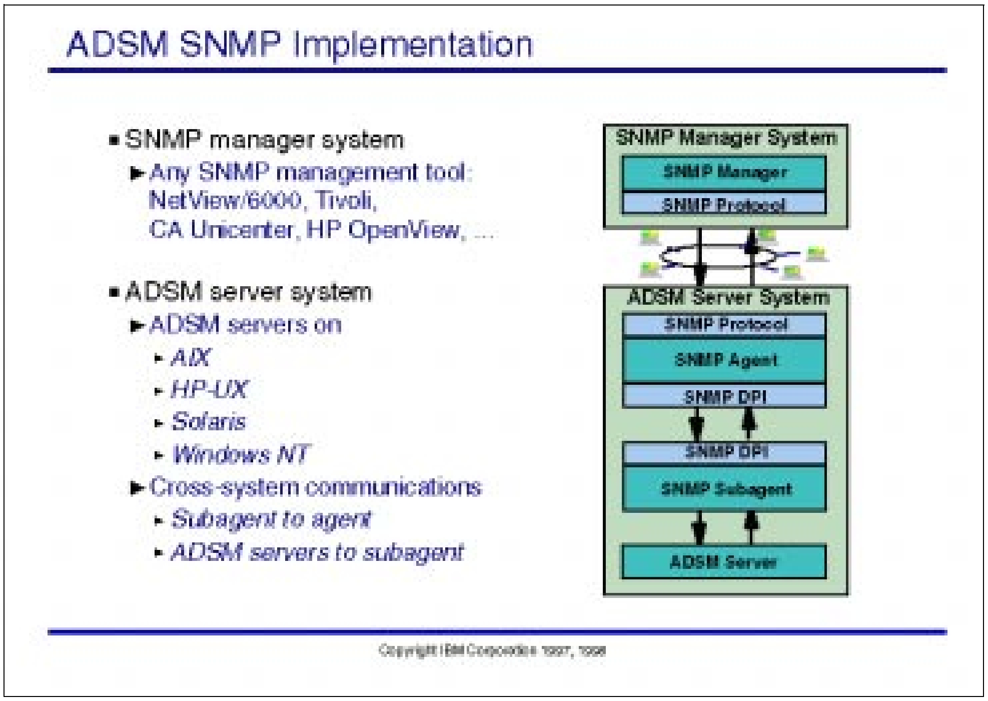
It is the agent's responsibility to answer the manager's requests and to inform the manager about events by sending traps. The agent communicates with both the manager and subagent. It can send queries to the subagent and receives traps that inform the SNMP manager about events taking place on the application monitored through the subagent.

9.3.3.4 SNMP Subagent

As the SNMP agent is responsible for answering manager requests and sending traps to the manager, the subagent answers MIB queries of the agent and informs the agent about events by sending traps. Additionally the subagent can create or delete objects or subtrees in the agent's MIB. This mechanism allows the subagent to define to the agent all the information required to monitor the managed application.

9.3.3.5 Agent-Subagent Communication

The SNMP agent and subagent communicate through the DPI. Communication takes place over a stream connection, which typically is a TCP/IP connection but could be another stream-oriented transport mechanism.



ADSM SNMP Implementation

- SNMP manager system
  - Any SNMP management tool: NetView/6000, Tivoli, CA Unicenter, HP OpenView, ...
- ADSM server system
  - ADSM servers on
    - *AIX*
    - *HP-UX*
    - *Solaris*
    - *Windows NT*
  - Cross-system communications
    - *Subagent to agent*
    - *ADSM servers to subagent*

Copyright IBM Corporation 1997, 1998

9.3.4 ADSM SNMP Implementation

To enable ADSM for SNMP, a subagent has been implemented in ADSM Version 3. To use this subagent, an SNMP manager is required and the ADSM server must be configured.

9.3.4.1 SNMP Manager System

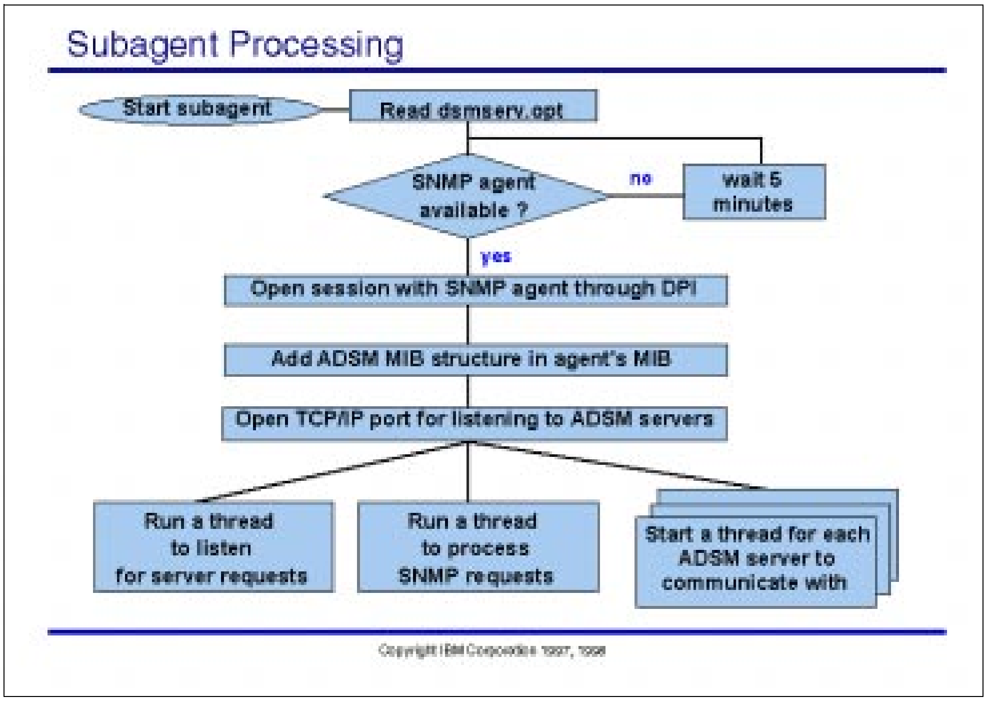
The SNMP manager system can reside on the same system as the ADSM server, but typically would be on another system connected through SNMP. The SNMP management tool can be any application that can manage information through SNMP MIB monitoring and traps. Such applications are NetView/6000, Tivoli, CA Unicenter, and HP OpenView.

9.3.4.2 ADSM Server System

ADSM monitoring through SNMP is supported for ADSM servers on AIX, HP-UX, Solaris, and Windows NT. There is no SNMP support for ADSM on MVS or OS/400. The ADSM server system runs the processes that are necessary to send ADSM event information to an SNMP management system. The processes are:

- SNMP agent (snmpd)
- ADSM SNMP subagent (dsmsnmp)
- ADSM server (dsmserv)

Cross-system support is available for subagent-to-agent and server-to-subagent communications. Thus an ADSM SNMP subagent can be used in managing multiple ADSM servers on separate hosts.



Subagent Processing

Start subagent

Read dsmserv.opt

SNMP agent available ?

no

wait 5 minutes

yes

Open session with SNMP agent through DPI

Add ADSM MIB structure in agent's MIB

Open TCP/IP port for listening to ADSM servers

Run a thread to listen for server requests

Run a thread to process SNMP requests

Start a thread for each ADSM server to communicate with

Copyright IBM Corporation 1997, 1998

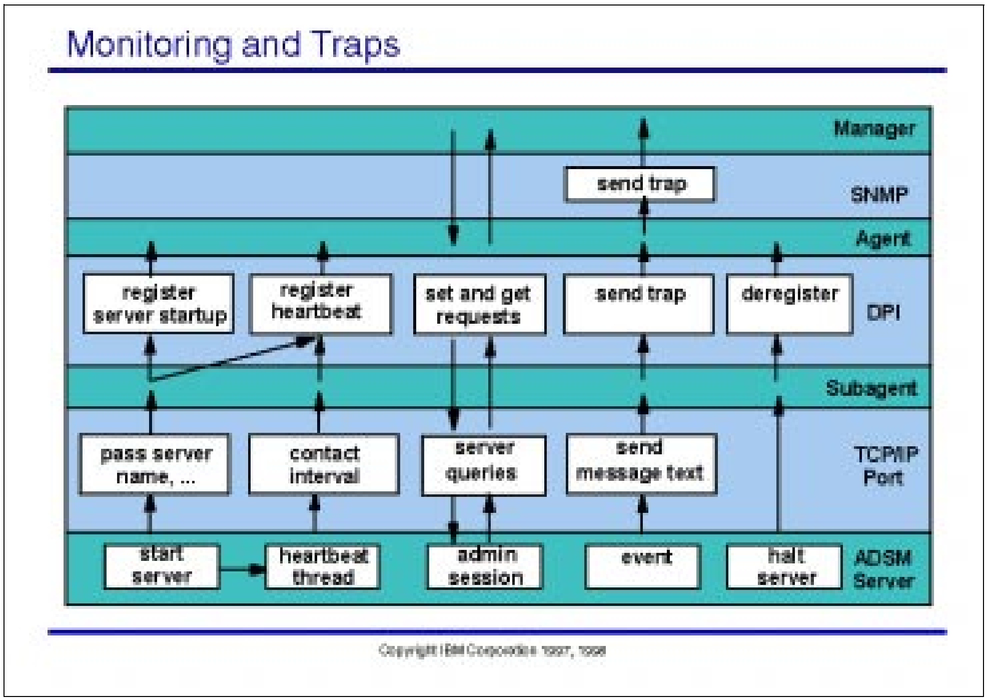
9.3.5 Subagent Processing

The ADSM SNMP subagent is started as a separate process called DSMSNMP. This executable file is located in the ADSM server directory. When started, the subagent reads the ADSM SNMP configuration options in dsmserv.opt. Because server and subagent share the same options file, they both know the TCP/IP information that the subagent uses to listen for server requests. The subagent determines whether an SNMP agent is available for communication. If an SNMP agent is not found, the subagent checks every 5 minutes for the presence of an agent.

As soon as an agent is found, the subagent performs the following steps:

1. Opens a session with the SNMP agent through the DPI
2. Registers the ADSM subtree in the agent's MIB through the DPI
3. Initializes the ADSM MIB structure representation in the agent's MIB through the DPI
4. Opens the configured TCP/IP port for listening to the ADSM server
5. Runs a thread to listen for ADSM server requests
6. Runs a thread to process SNMP requests through the DPI
7. Runs a thread to communicate with each ADSM server that has registered

When an ADSM server session terminates, the server is automatically deregistered.



Monitoring and Traps

Copyright IBM Corporation 1997, 1998

9.3.6 Monitoring and Traps

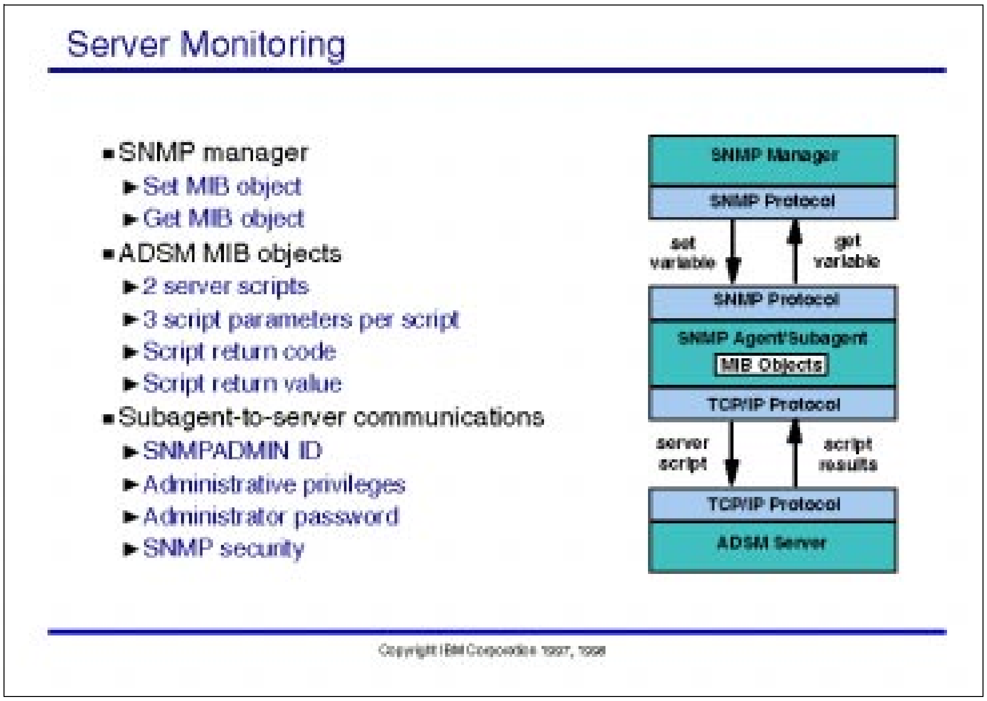
When an ADSM server starts up, it passes its server name, starting directory, and server level to the ADSM subagent. The contact time is recorded in the agent's MIB.

An ADSM server heartbeat thread contacts the subagent within a configured time interval. Every time the subagent is contacted by this heartbeat thread, it records the time in the agent's MIB.

The ADSM subagent contains additional MIB objects that are used for monitoring an ADSM server. These objects can be modified and read by an SNMP manager issuing *set* and *get* requests for the objects. The MIB objects are used to execute server scripts on the ADSM server.

When the ADSM server encounters an event and the event has been enabled for the SNMP receiver, the server passes the message to the subagent. The subagent sends this information as a trap to the agent, which the agent then sends as a trap to the SNMP manager.

When an ADSM server is shut down, the server-related information is deregistered in the agent's MIB.



Server Monitoring

- SNMP manager
  - Set MIB object
  - Get MIB object
- ADSM MIB objects
  - 2 server scripts
  - 3 script parameters per script
  - Script return code
  - Script return value
- Subagent-to-server communications
  - SNMPADMIN ID
  - Administrative privileges
  - Administrator password
  - SNMP security

9.3.7 Server Monitoring

SNMP monitoring enables an SNMP manager to actively monitor an ADSM server by executing server scripts that are defined on the server. Only scripts already defined on the server can be executed in this manner.

9.3.7.1 SNMP Manager

An SNMP manager performs operations on a managed system by modifying and reading MIB objects defined in the SNMP agent and subagent. MIB objects are variables whose values can be modified by issuing an SNMP *set* command with the name of the MIB object and the value to which to set it. MIB objects are read from the SNMP agent and subagent through an SNMP *get* command.

9.3.7.2 ADSM MIB Objects

The ADSM MIB contains a number of MIB objects that are used to execute server scripts on an ADSM server, pass script parameters, and retrieve the script results. Two MIB objects are defined for server script names:

- ibmAdsmServerScript1
- ibmAdsmServerScript2

Both of these MIB objects can have the name of a server script written into them by an SNMP manager issuing an SNMP set command. Each script MIB object has six associated MIB objects:

1. ibmAdsmServerM***x***Parm1
2. ibmAdsmServerM***x***Parm2
3. ibmAdsmServerM***x***Parm3
4. ibmAdsmServerM***x***ReturnLength
5. ibmAdsmServerM***x***ReturnCode
6. ibmAdsmServerM***x***ReturnValue

The associated MIB objects for the two scripts have similar names, with the *x* value of 1 for Script1 or 2 for Script2.

The Parm1, Parm2, and Parm3 MIB objects are used to pass a maximum of three parameters for use when the script is executed. Again, these values are written by an SNMP manager issuing an SNMP set command for that object name. Use of the Parm*x* MIB objects is optional; a script can be used without parameters. These MIB objects are rewritable from an SNMP manager and can be set to different values as required. However, only two script names can be used concurrently.

The ReturnLength, ReturnCode, and ReturnValue MIB objects contain information returned from the ADSM server when the script is executed. Once a script name has been set in either the Script1 or Script2 MIB object, it is executed by the SNMP manager performing an SNMP *get* command for the associated ReturnValue MIB option. The get command causes the script to be executed on the ADSM server. The output from the script execution, which contains carriage return and line feed characters to ensure correct formatting, is written into the ReturnValue MIB object and then sent to the SNMP manager. The SNMP manager or an MIB browser used to view the ReturnValue MIB must be able to handle embedded carriage return and line feed characters.

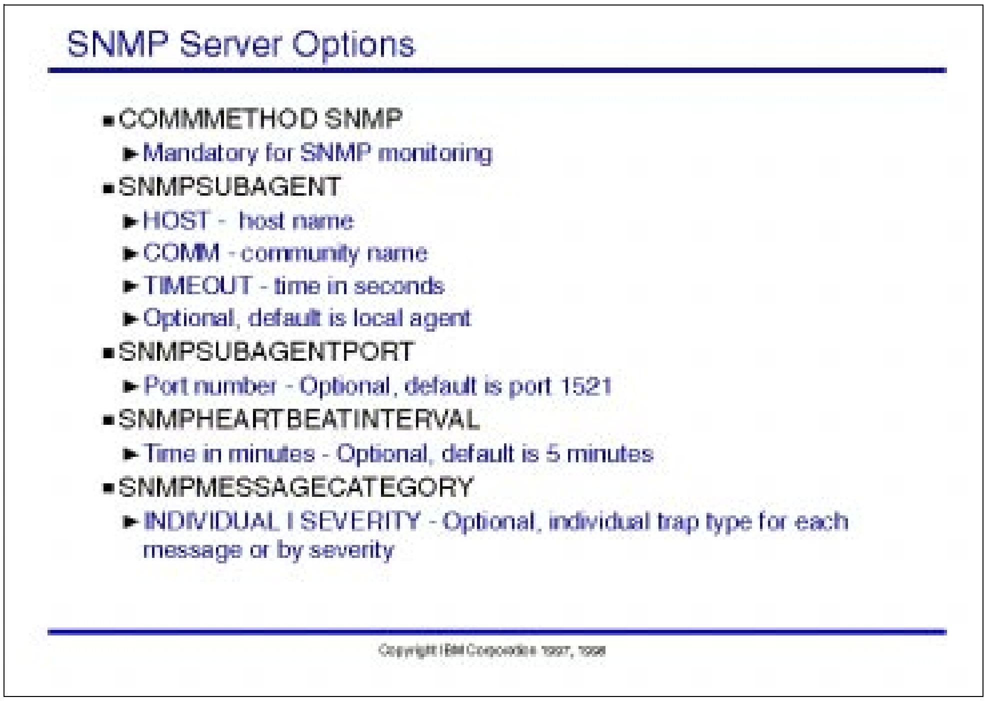
9.3.7.3 Subagent-to-Server Communications

A modified type of administrative session is used between the SNMP subagent and server to execute server scripts. When the ADSM server starts up and registers with the SNMP subagent, it passes to the subagent the TCP/IP port number on which the server is listening for incoming client connections. The subagent uses this port number to contact the server when a script has to be executed. The subagent starts an administrative client session, using an SNMPADMIN administrative ID. This session is used to execute the script and return the results to the subagent. No authentication is performed by the server when the subagent uses SNMPADMIN ID to start such a session.

The SNMPADMIN administrative ID is not automatically defined on the server. It must be defined by an administrator before SNMP monitoring can be performed. Although the subagent does not use authentication, a password and administrative privileges should be set for this ID. The administrative password prevents unauthorized use of this ID by other persons. An appropriate level of administrative privilege should also be assigned for the type of monitoring required. For example, assigning only analyst privileges to SNMPADMIN will prevent any unauthorized user of this ID from making updates to the server.

To perform server monitoring, the SNMP agent must be configured to give the SNMP manager readWrite authority to the ADSM MIB objects (see “SNMP Setup” on page 333). This enables the SNMP manager to issue *set* commands for the MIB objects.

Access to the SNMP agent is controlled by *community names* and *passwords* defined on the SNMP agent. These are transmitted across the network as plain text. This is a potential security exposure that could allow unauthorized access to the MIB objects. Setting an appropriate administrative authority for the SNMPADMIN ID on the server does not resolve this SNMP security issue, but it will limit unauthorized users from making updates to the ADSM server.



SNMP Server Options

- COMMMETHOD SNMP
  - Mandatory for SNMP monitoring
- SNMPSUBAGENT
  - HOST - host name
  - COMM - community name
  - TIMEOUT - time in seconds
  - Optional, default is local agent
- SNMPSUBAGENTPORT
  - Port number - Optional, default is port 1521
- SNMPHEARTBEATINTERVAL
  - Time in minutes - Optional, default is 5 minutes
- SNMPMESSAGECATEGORY
  - INDIVIDUAL | SEVERITY - Optional, individual trap type for each message or by severity

9.3.8 SNMP Server Options

The following options can be specified in dsmserv.opt to configure communication between the ADSM server and ADSM subagent.

9.3.8.1 COMMETHOD SNMP

A COMMETHOD SNMP option is mandatory to enable monitoring through SNMP.

9.3.8.2 SNMPSUBAGENT

SNMPSUBAGENT has three parameters:

- HOST
- COMM
- TIMEOUT

where HOST specifies the TCP/IP number of the host running the SNMP agent (to which the subagent will connect). COMM specifies the same community name as the configured community name on the system running the SNMP agent. TIMEOUT specifies the time in seconds in which a request must be received. These parameters are optional and default to the local agent. For server monitoring, the TIMEOUT value may have to be adjusted to prevent timeouts from occurring while long-running server scripts are executing.

9.3.8.3 SNMPSUBAGENTPORT

SNMPSUBAGENTPORT is optional. It specifies the TCP/IP port number for the ADSM server-to-subagent communication.

9.3.8.4 SNMPHEARTBEATINTERVAL

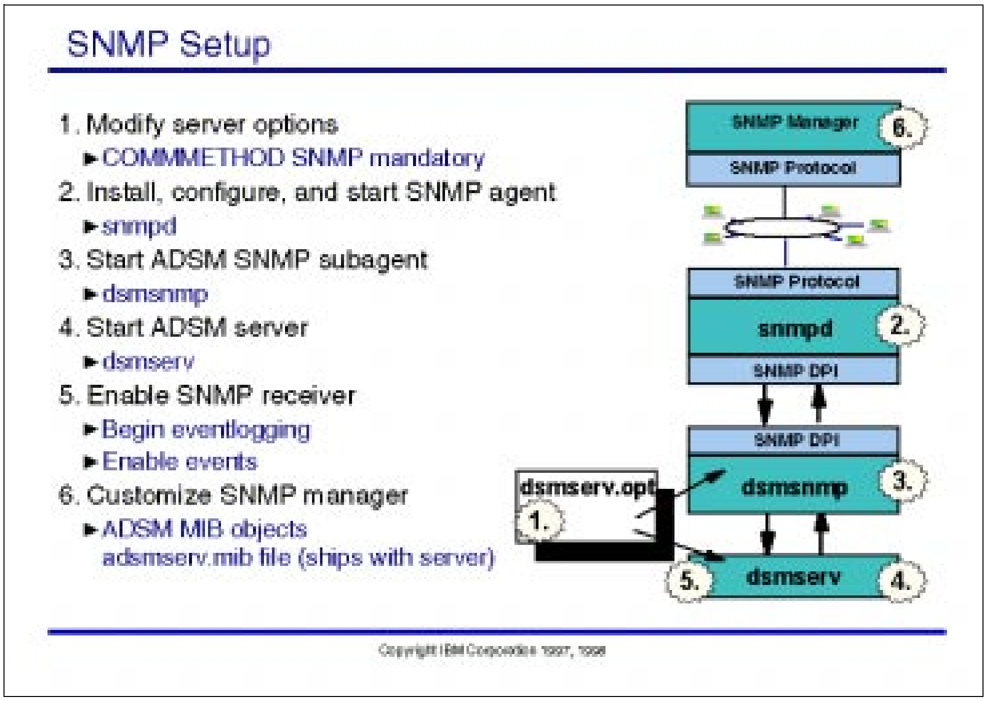
SNMPHEARTBEATINTERVAL specifies in minutes the interval in which the server heartbeat contacts the subagent to indicate that it is still alive. This parameter is optional. If not specified, the default value is 5 minutes.

9.3.8.5 SNMPMESSAGECATEGORY

SNMPMESSAGECATEGORY can be set to INDIVIDUAL or SEVERITY. INDIVIDUAL causes the subagent to assign a separate trap type for each event message number forwarded from the server to the SNMP manager. SEVERITY sends events as one of four trap types on the basis of the ADSM message severity: Severe, Error, Warning, or Information. SEVERITY is the default.

An SNMP configuration in dsmserv.opt might look like this:

```
commmethod             snmp
snmpsubagent           host 9.1.150.69 comm public timeout 120
snmpsubagentport       1521
snmpheartbeatinterval  5
snmpmessagecategory    severity
```



SNMP Setup

1. Modify server options
   - COMMMETHOD SNMP mandatory
2. Install, configure, and start SNMP agent
   - snmpd
3. Start ADSM SNMP subagent
   - dsmsnmp
4. Start ADSM server
   - dsmserv
5. Enable SNMP receiver
   - Begin eventlogging
   - Enable events
6. Customize SNMP manager
   - ADSM MIB objects
     adsmserv.mib file (ships with server)

Copyright IBM Corporation 1997, 1998

9.3.9 SNMP Setup

To set up ADSM monitoring through SNMP, perform the following:

1. Modify server options

   The dsmserv.opt must include the entry COMMMETHOD SNMP in order to communicate through SNMP. Further SNMP options are optional (see “SNMP Server Options” on page 331 for explanation of SNMP parameters). Both the ADSM server and ADSM subagent use the dsmserv.opt file. Therefore configuration of this file has to take place first.

2. Install, configure, and start SNMP agent

   The installed SNMP agent must support the DPI Version 2.0 standard (See “SNMP Agent” on page 322). The agent's configuration is described in the documentation delivered with the agent's software. For example, the AIX SystemView agent is configured by customizing the file /etc/snmpd.conf. A default configuration might look like this:

```
logging     file=/var/snmp/snmpd.log  enabled
logging     size=0  level=0

community   public
community   private 127.0.0.1   255.255.255.255 readWrite
community   system  127.0.0.1   255.255.255.255 readWrite

view        1.17.2 system enterprises view

trap        public  snmp_manager_ip_adr 1.2.3 fe

snmpd       maxpacket=16000 smuxtimeout=60
```

```
smux        1.3.6.1.4.1.2.3.1.2.1.2   gated_password

smux        1.3.6.1.4.1.2.3.1.2.2.1.1.2  dpid_password
```

where **snmp_manager_ip_adr** has to be substituted by the IP address of the system running the SNMP management application. The community names specified must also grant **readWrite** authority to the MIB if server monitoring is to be used (see “Server Monitoring” on page 329).

Before starting the agent, you must ensure that the DPI agent will be started and not the default SNMP agent shipping with the operating system or TCP/IP. If using the SystemView agent, you must set the SVA_SNMPD environment variable to ensure that the correct agent is started. You can set the variable to any value. For example, on AIX (korn shell) use the following export command:

```
# export SVA_SNMPD="active"
```

3. Start ADSM server

   After you have activated the agent and subagent, start the ADSM server to begin communication through the configured TCP/IP port with the subagent.

4. Enable SNMP receiver

   To enable events for the SNMP receiver issue the BEGIN EVENTLOGGING command for the SNMP receiver. Then issue the ENABLE EVENT command for individual events or event severities to be reported to SNMP:

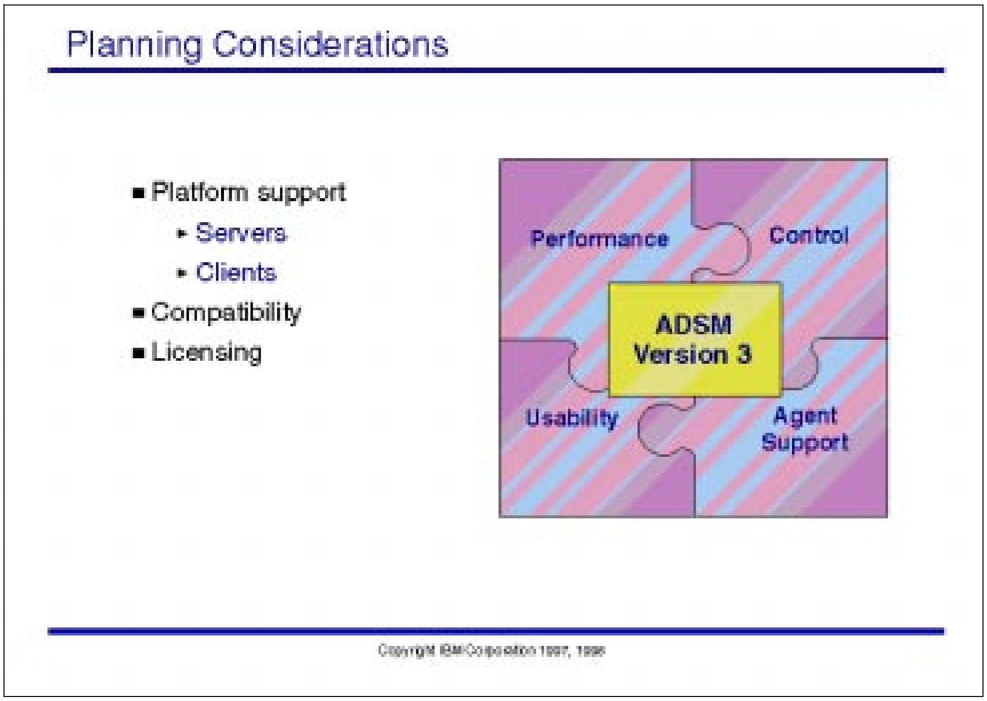
   ```
   adsm> begin eventlogging snmp

   adsm> enable event snmp all
   ```

5. Customize SNMP manager

   Define the ADSM SNMP MIB values for the SNMP manager to help format and display the ADSM SNMP MIB variables and messages. The ADSMSERV.MIB file ships with the ADSM server and must be loaded by the SNMP manager. For example, when you run NetView for OS/2 as an SNMP manager, the ADSMSERV.MIB file is copied to the \netview_path\SNMP_MIB directory and then loaded through this command:

   ```
   [C:\] loadmib -load adsmserv.mib
   ```

Chapter 10. Planning Considerations



Planning Considerations

- Platform support
  - Servers
  - Clients
- Compatibility
- Licensing

Performance

Control

ADSM Version 3

Usability

Agent Support

Copyright IBM Corporation 1997, 1998

This chapter contains planning information related to implementing or upgrading to ADSM Version 3:

- Platform support

  This section details the supported platforms for ADSM Version 3 servers and clients.

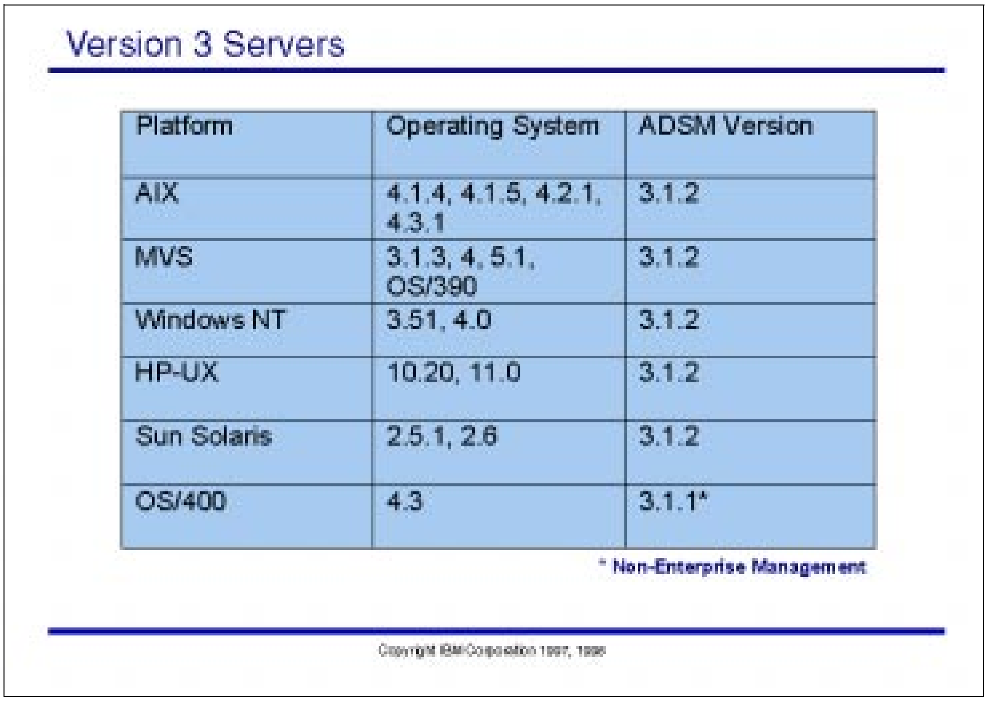
- Compatibility

  This section details the new functions available in a mixed environment of Version 2 and 3 clients and servers.

- Licensing and packaging

  This section explains the licensing structure for the AIX, MVS, and Windows NT servers.

10.1 Platform Support



Version 3 Servers

Platform	Operating System	ADSM Version
AIX	4.1.4, 4.1.5, 4.2.1, 4.3.1	3.1.2
MVS	3.1.3, 4, 5.1, OS/390	3.1.2
Windows NT	3.51, 4.0	3.1.2
HP-UX	10.20, 11.0	3.1.2
Sun Solaris	2.5.1, 2.6	3.1.2
OS/400	4.3	3.1.1*

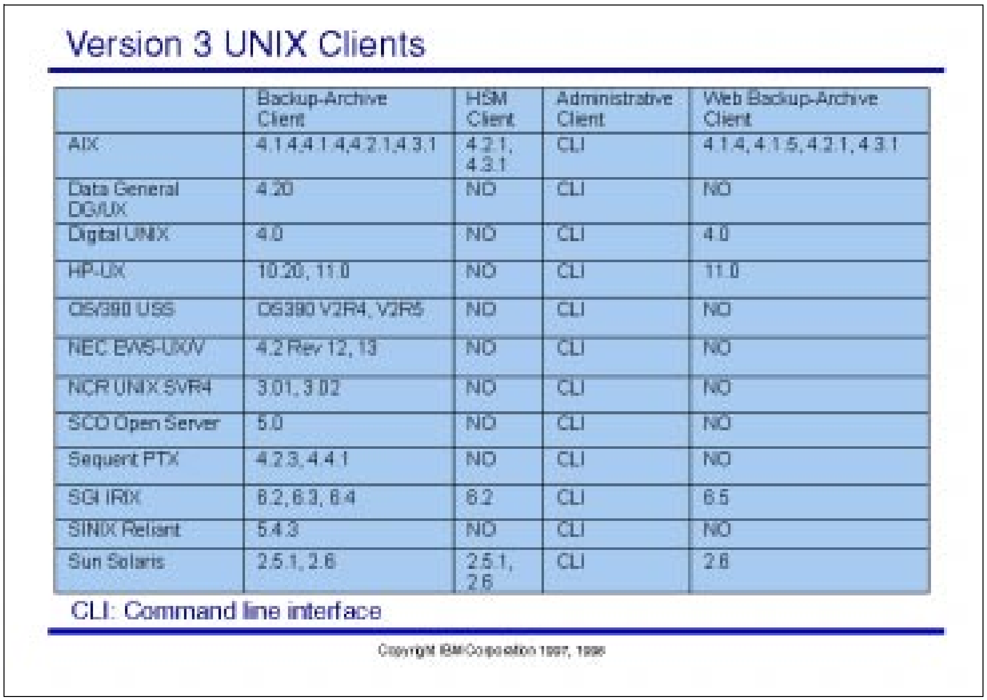
* Non-Enterprise Management

10.1.1 Version 3 Servers

All ADSM Version 3 servers, with the exception of the OS/400 server, support the Enterprise Management enhancements shipped with ADSM Version 3.1.2 This graphic lists the supported operating system releases and availability dates for the Version 3 servers.

Note

The Microsoft Windows NT server is for the Intel platform only.



Version 3 UNIX Clients

	Backup-Archive Client	HSM Client	Administrative Client	Web Backup-Archive Client
AIX	4.1.4,4.1.4,4.2.1,4.3.1	4.2.1, 4.3.1	CLI	4.1.4, 4.1.5, 4.2.1, 4.3.1
Data General DG/UX	4.20	NO	CLI	NO
Digital UNIX	4.0	NO	CLI	4.0
HP-UX	10.20, 11.0	NO	CLI	11.0
OS/390 USS	OS390 V2R4, V2R5	NO	CLI	NO
NEC EWS-UX/V	4.2 Rev 12, 13	NO	CLI	NO
NCR UNIX SVR4	3.01, 3.02	NO	CLI	NO
SCO Open Server	5.0	NO	CLI	NO
Sequent PTX	4.2.3, 4.4.1	NO	CLI	NO
SGI IRIX	6.2, 6.3, 6.4	6.2	CLI	6.5
SINIX Reliant	5.4.3	NO	CLI	NO
Sun Solaris	2.5.1, 2.6	2.5.1, 2.6	CLI	2.6

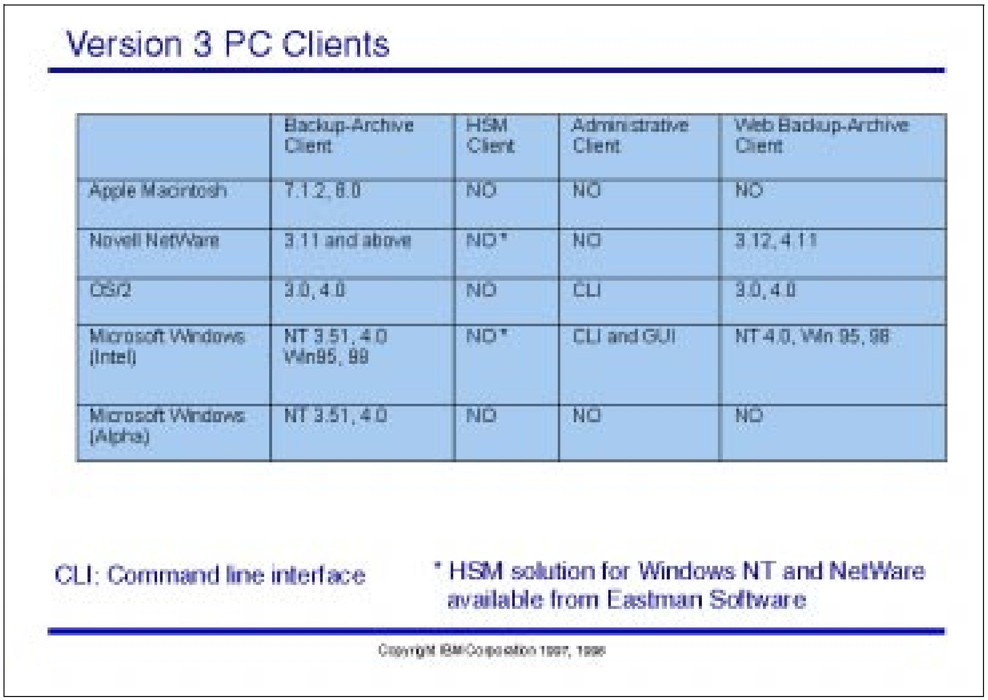
CLI: Command line interface

10.1.2 Version 3 UNIX Clients

This graphic details the UNIX operating systems that are supported as clients by ADSM Version 3. Only the minimum levels for the various client components are shown. There are no Version 3 clients for the following platforms:

- Auspex
- Bull
- Digital ULTRIX
- Pyramid Nile
- SunOS

These five platforms, supported in Version 2, can continue to use their existing Version 2 clients. All existing Version 2 clients are supported with Version 3 servers. Version 1 clients are not supported with Version 3 servers.



Version 3 PC Clients

	Backup-Archive Client	HSM Client	Administrative Client	Web Backup-Archive Client
Apple Macintosh	7.1.2, 8.0	NO	NO	NO
Novell NetWare	3.11 and above	NO *	NO	3.12, 4.11
OS/2	3.0, 4.0	NO	CLI	3.0, 4.0
Microsoft Windows (Intel)	NT 3.51, 4.0 Win95, 98	NO *	CLI and GUI	NT 4.0, Win 95, 98
Microsoft Windows (Alpha)	NT 3.51, 4.0	NO	NO	NO

CLI: Command line interface

* HSM solution for Windows NT and NetWare available from Eastman Software

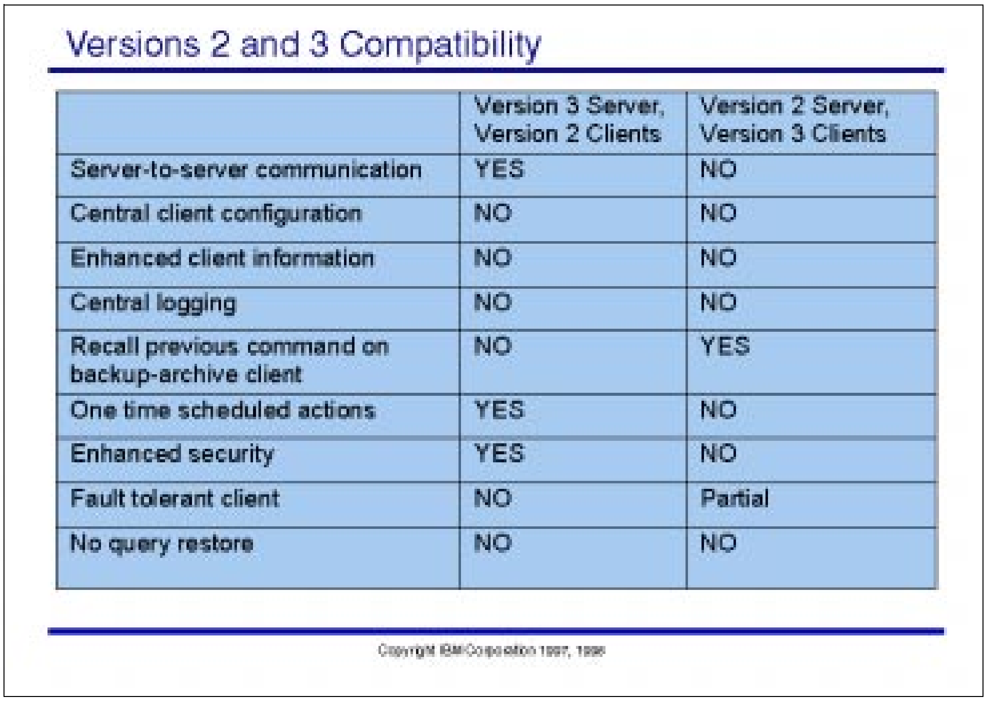
10.1.3 Version 3 PC Clients

The graphic details the PC operating systems that are be supported as clients by ADSM Version 3. Only the minimum levels for the various client components are shown. There are no Version 3 backup-archive clients for the following platforms:

- DOS
- Microsoft Windows 3.1

These two platforms, supported in Version 2, can continue to use their existing Version 2 clients. All existing Version 2 clients are supported with Version 3 servers. Version 1 clients are not supported with Version 3 servers.

10.2 Versions 2 and 3 Compatibility

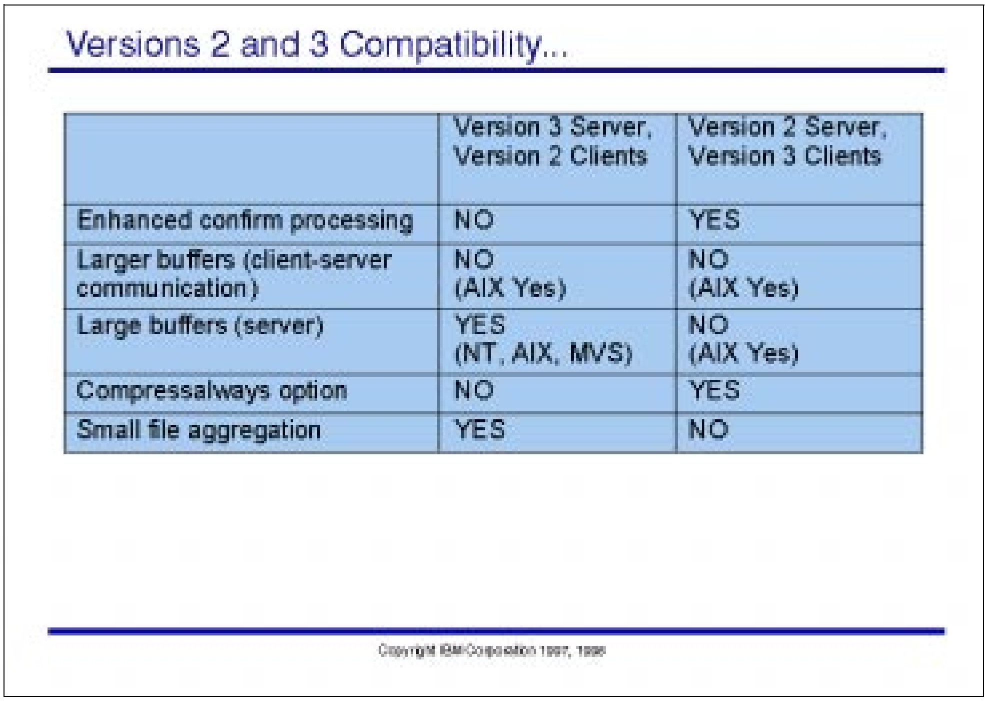


Versions 2 and 3 Compatibility

	Version 3 Server, Version 2 Clients	Version 2 Server, Version 3 Clients
Server-to-server communication	YES	NO
Central client configuration	NO	NO
Enhanced client information	NO	NO
Central logging	NO	NO
Recall previous command on backup-archive client	NO	YES
One time scheduled actions	YES	NO
Enhanced security	YES	NO
Fault tolerant client	NO	Partial
No query restore	NO	NO

10.2.1 Versions 2 and 3 Compatibility

Version 3 clients are supported with Version 2 servers. This graphic details the combination of client and server versions that support the Version 3 functions.

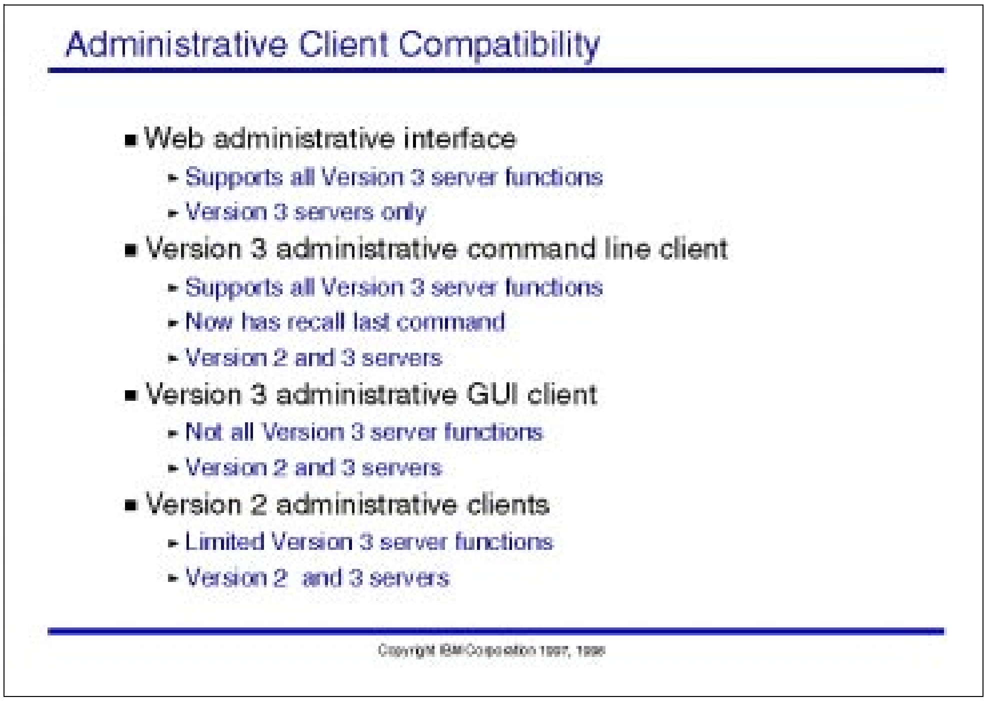


Versions 2 and 3 Compatibility...

	Version 3 Server, Version 2 Clients	Version 2 Server, Version 3 Clients
Enhanced confirm processing	NO	YES
Larger buffers (client-server communication)	NO (AIX Yes)	NO (AIX Yes)
Large buffers (server)	YES (NT, AIX, MVS)	NO (AIX Yes)
Compressalways option	NO	YES
Small file aggregation	YES	NO

10.2.2 Versions 2 and 3 Compatibility...

Version 3 clients are supported with Version 2 servers. This graphic details the combination of client and server versions that support the Version 3 functions.



Administrative Client Compatibility

- Web administrative interface
  - Supports all Version 3 server functions
  - Version 3 servers only
- Version 3 administrative command line client
  - Supports all Version 3 server functions
  - Now has recall last command
  - Version 2 and 3 servers
- Version 3 administrative GUI client
  - Not all Version 3 server functions
  - Version 2 and 3 servers
- Version 2 administrative clients
  - Limited Version 3 server functions
  - Version 2 and 3 servers

Copyright IBM Corporation 1997, 1998

10.2.3 Administrative Client Compatibility

Version 3 introduces significant changes to the administrative interfaces.

10.2.3.1 Web Administrative Interface

The Web administrative interface is the new interface for administering Version 3 servers. It supports all server operations through a GUI and command line screen. It has the best platform support as it can be used with any Web browser that supports HTML 2.0 or higher.

10.2.3.2 Version 3 Administrative Command Line Client

The Version 3 administrative command line client supports all server operations. It has been enhanced to recall previous commands.

It is available on all backup-archive platforms except Novell NetWare and Apple Macintosh and works with either a Version 2 or 3 server.

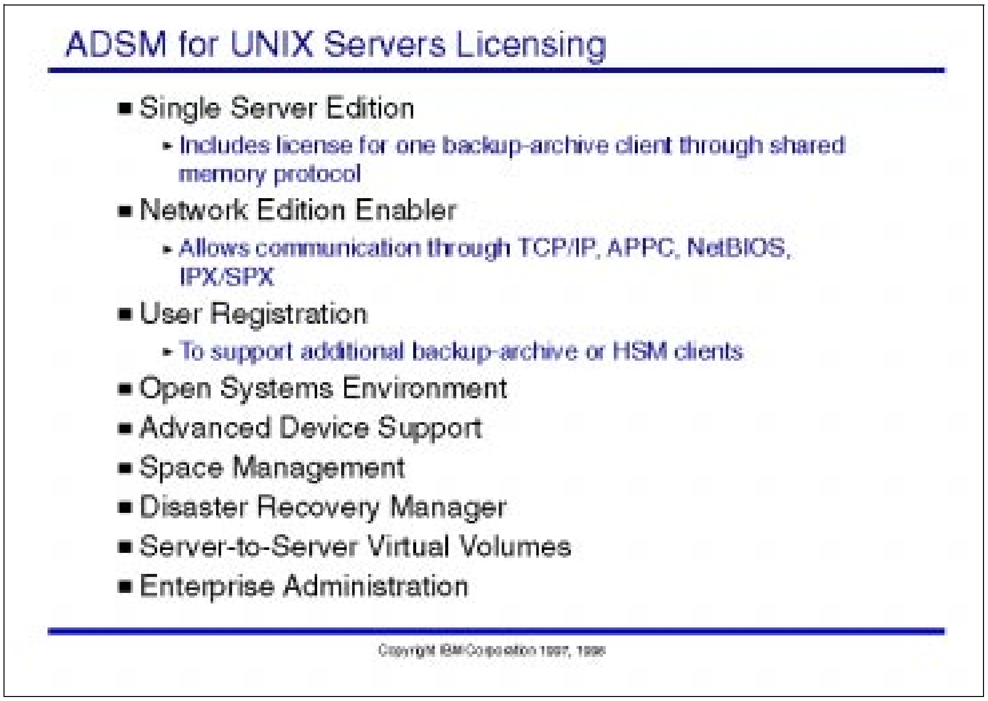
10.2.3.3 Version 3 Administrative GUI Client

The Version 3 administrative GUI client is supported on Windows 95 and Windows NT only. It supports fewer operations than the Web administrative interface. It can be used with Version 2 and Version 3 servers.

10.2.3.4 Version 2 Administrative Clients

The Version 2 administrative GUI client supports most, but not all, Version 2 server operations. They can be used with Version 2 and Version 3 servers.

10.3 Licensing and Packaging



ADSM for UNIX Servers Licensing

- Single Server Edition
  - Includes license for one backup-archive client through shared memory protocol
- Network Edition Enabler
  - Allows communication through TCP/IP, APPC, NetBIOS, IPX/SPX
- User Registration
  - To support additional backup-archive or HSM clients
- Open Systems Environment
- Advanced Device Support
- Space Management
- Disaster Recovery Manager
- Server-to-Server Virtual Volumes
- Enterprise Administration

10.3.1 ADSM for UNIX Servers Licensing

The licensing is the same for the ADSM servers on the AIX, HP-UX, and Sun Solaris platforms.

ADSM for AIX Version 3 is product number 5765-C43. ADSM for HP-UX Version 3 is product number 5639-D92. ADSM for Sun Solaris Version 3 is product number 5639-D91.

10.3.1.1 Single Server Edition

The Single Server Edition includes a license for one backup-archive client through the shared memory protocol, that is, on the same machine as the server only, and for administrators connected to the server by any method.

10.3.1.2 Network Edition Enabler

The Network Edition Enabler allows communication through network protocols.

10.3.1.3 User Registration

A User Registration licenses one backup-archive and/or HSM client.

10.3.1.4 Open Systems Environment

The Open Systems license is required for AFS/DFS clients and/or MVS/OE clients. If both an MVS/OE client and a DFS client are managed, this license is needed only once. Given that AFS, DFS, or MVS/OE clients must be network attached, the Network Edition Enabler is also required.

10.3.1.5 Advanced Device Support

The Extended Device Support licenses high-end devices, for example, those that were in Device Modules 3 or 4 for ADSM Version 2.

10.3.1.6 Space Management

The space management license is required for HSM clients. Machines running HSM clients must have a User Registration license. However, a machine that runs both HSM and the backup-archive client requires only one User Registration license.

10.3.1.7 Disaster Recovery Manager

The DRM license is required for Disaster Recovery Manager.

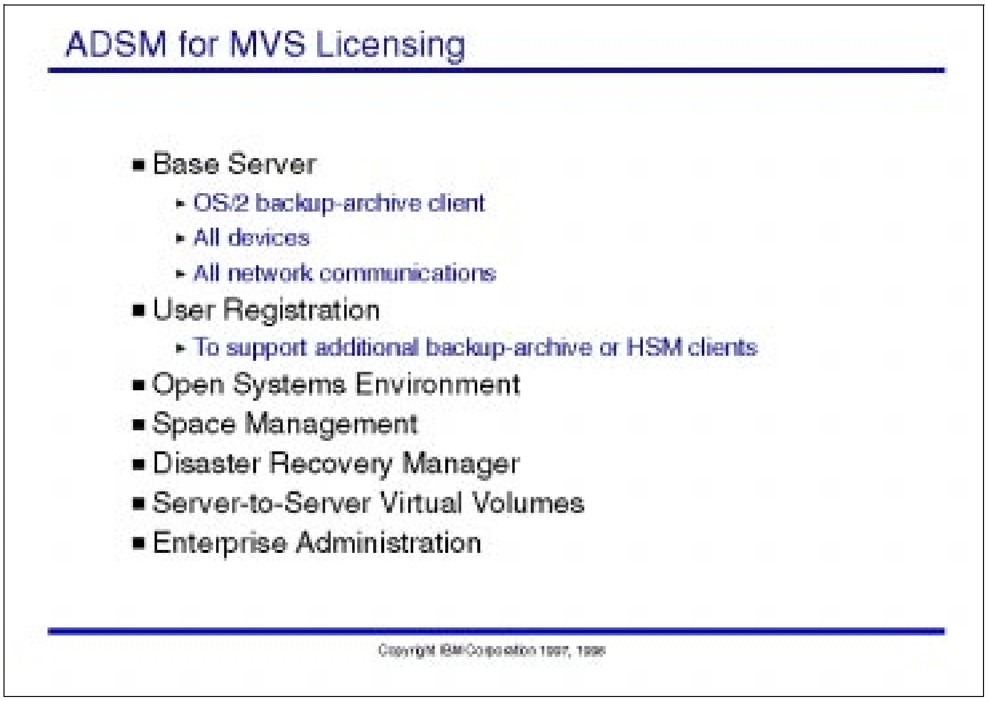
10.3.1.8 Server-to-Server Virtual Volumes

This license is required if the server uses server-to-server virtual volumes.

10.3.1.9 Enterprise Administration

This license is required for the following functions:

- Administrative command routing
- Enterprise configuration
- Server-to-server event logging



ADSM for MVS Licensing

- Base Server
  - OS/2 backup-archive client
  - All devices
  - All network communications
- User Registration
  - To support additional backup-archive or HSM clients
- Open Systems Environment
- Space Management
- Disaster Recovery Manager
- Server-to-Server Virtual Volumes
- Enterprise Administration

10.3.2 ADSM for MVS Licensing

ADSM for MVS Version 3 is product number 5655-A30.

10.3.2.1 Base Server

The Base Server includes license for one backup-archive client, license for any supported devices, and communication to clients over any network protocol. Code for a TSO administrative command line client is included.

10.3.2.2 User Registration

A User Registration licenses one backup-archive and/or HSM client

10.3.2.3 Open Systems Environment

The Open Systems license is required for AFS/DFS clients and/or MVS/OE clients. If both an MVS/OE client and a DFS client are managed, this license is needed only once.

10.3.2.4 Space Management

The space management license is required for HSM clients. Machines running HSM clients must have a User Registration license. However, a machine that runs both HSM and the backup-archive client requires only one User Registration license.

10.3.2.5 Disaster Recovery Manager

The DRM license is required for Disaster Recovery Manager.

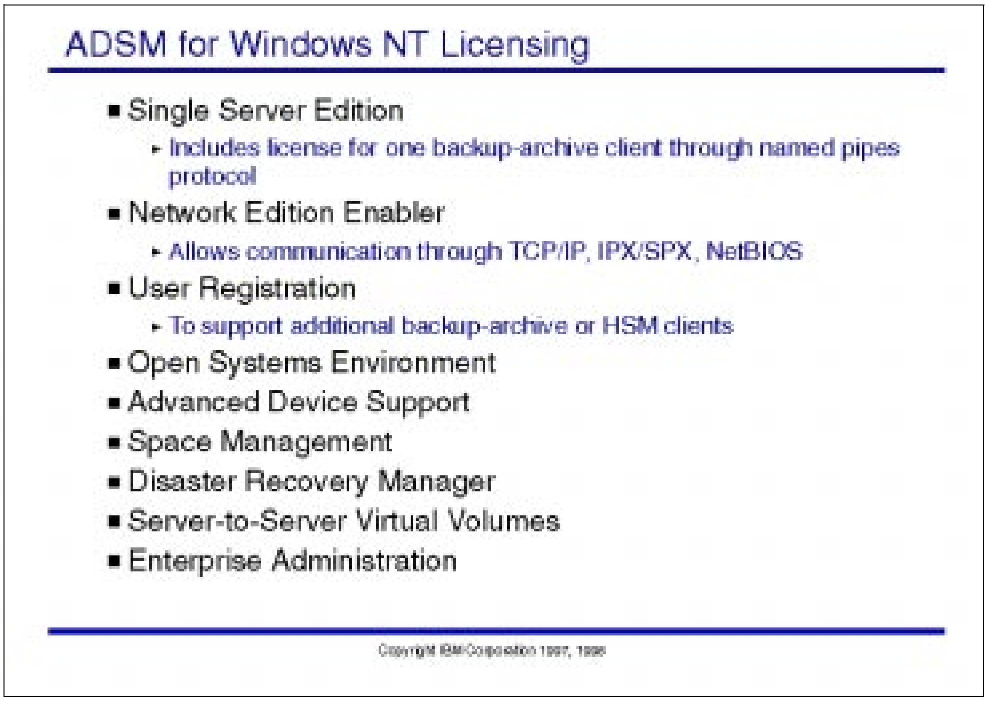
10.3.2.6 Server-to-Server Virtual Volumes

This license is required if the server uses server-to-server virtual volumes.

10.3.2.7 Enterprise Administration

This license is required for the following functions:

- Administrative command routing
- Enterprise configuration
- Server-to-server event logging



ADSM for Windows NT Licensing

- Single Server Edition
  - Includes license for one backup-archive client through named pipes protocol
- Network Edition Enabler
  - Allows communication through TCP/IP, IPX/SPX, NetBIOS
- User Registration
  - To support additional backup-archive or HSM clients
- Open Systems Environment
- Advanced Device Support
- Space Management
- Disaster Recovery Manager
- Server-to-Server Virtual Volumes
- Enterprise Administration

Copyright IBM Corporation 1997, 1998

10.3.3 ADSM for Windows NT Licensing

ADSM for Windows NT Version 3 is product number 5639-C59.

10.3.3.1 Single Server Edition

The Single Server Edition includes license for one backup-archive client through named pipes protocol, that is, on the same machine as the server only, and for administrators connected to the server by any method.

10.3.3.2 Network Edition Enabler

The Network Edition Enabler allows communication through network protocols.

10.3.3.3 User Registration

A User Registration licenses one backup-archive and/or HSM client. A User Registration should be purchased for each backup-archive and HSM client. A machine running both backup-archive and HSM clients requires only one User Registration to be licensed at the ADSM server.

10.3.3.4 Open Systems Environment

The Open Systems license is required for AFS/DFS clients and MVS/OE clients. It is purchased only once, no matter how many AFS, DFS, and/or MVS/OE clients are managed. A User Registration is required for each of these clients. Given that AFS or DFS clients must be network attached, the Network Edition Enabler is also required.

10.3.3.5 Extended Device Support

The Extended Device Support licenses the same devices that were in the ADSM for Windows NT Advanced Device Support module.

10.3.3.6 Space Management

The space management license is required for HSM clients. Machines running HSM clients must have a User Registration license. However, a machine that runs both HSM and the backup-archive client requires only one User Registration license.

10.3.3.7 Disaster Recovery Manager

The DRM license is required for Disaster Recovery Manager.

10.3.3.8 Server-to-Server Virtual Volumes

This license is required if the server uses server-to-server virtual volumes.

10.3.3.9 Enterprise Administration

This license is required for the following functions:

- Administrative command routing
- Enterprise configuration
- Server-to-server event logging

Appendix A. Special Notices

This publication is intended to help customers, consultants, Business Partners, and IBMers understand ADSM Version 3. The information in this publication is not intended as the specification of any programming interfaces that are provided by ADSM Version 3. See the PUBLICATIONS section of the IBM Programming Announcement for ADSM Version 3 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, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM ®
ADSTAR
AIXwindows
Enterprise Systems Architecture/390
NetView
OS/390
PowerPC
ACF/VTAM
AIX/6000
AS/400
MVS/ESA
OS/2
OS/400
VM/ESA

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix B. Related Publications

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

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see , “How to Get ITSO Redbooks” on page 355.

ADSM Redbooks

Book Title	Publication Number
General Topics	
ADSM Concepts	SG24-4877
ADSM Advanced Implementation Experiences	GG24-4221
Using ADSM Hierarchical Storage Management	SG24-4631
Client Disaster Recovery: Bare Metal Restore	SG24-4880
Specific Server Books	
Windows NT Backup and Recovery with ADSM	SG24-2231
ADSM Server-to-Server Implementation and Operation	SG24-5244
ADSM Server for Windows NT Configuration and Recovery Examples	SG24-4878
Getting Started with ADSM/6000	GG24-4421
ADSM for AIX: Advanced Topics	SG24-4601
AIX Tape Management	SG24-4705
ADSM/6000 on 9076 SP2	GG24-4499
ADSM for MVS: Recovery and Disaster Recovery	SG24-4537
ADSM for MVS: Using Tapes and Tape Libraries	SG24-4538
Getting Started with ADSM/2	GG24-4321
ADSM for OS/2: Advanced Topics	SG24-4740
Setting Up and Implementing ADSM/400	GG24-4460
ADSM/VSE Implementation Guide	SG24-4266
Specific Client Books	
Getting Started with ADSM NetWare Clients	GG24-4242
Getting Started with ADSM AIX Clients	GG24-4243
ADSM API Examples for OS/2 and Windows	SG24-2588
ADSM with Other Products	
A Practical Guide to Network Storage Manager	SG24-2242
Using ADSM to Back Up Databases	SG24-4335
Using ADSM to Back Up Lotus Notes	SG24-4534
Hierarchical Storage Management for NetWare: ADSM and AvailHSM Implementation	SG24-4713
Using ADSM to Back Up OS/2 LAN Server and Warp Server	SG24-4682
Backup, Recovery, and Availability with DB2 Parallel Edition on RISC/6000	SG24-4695
ADSM Operation and Management with TME10	SG24-2214
ADSM Reporting with SAMS:Vantage	SG24-5271

B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	**Subscription Number**	**Collection Kit Number**
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

B.3 ADSM Product Publications

Book Title	**Publication Number**
ADSM V3R1 Messages	GC35-0271
ADSM V3R1 AIX Quick Start	GC35-0273
ADSM V3R1 AIX Administrator’s Guide	GC35-0274
ADSM V3R1 AIX Administrator’s Reference	GC35-0275
ADSM V3R1 MVS Quick Start	GC35-0276
ADSM V3R1 MVS Administrator’s Guide	GC35-0277
ADSM V3R1 MVS Administrator’s Reference	GC35-0278
ADSM V3R1 AIX License	SC35-0283
ADSM V3R1 MVS License	GC35-0284
ADSM V3R1 Windows NT Administrator’s Guide	GC35-0292
ADSM V3R1 Windows NT Administrator’s Reference	GC35-0293
ADSM V3R1 Windows NT License	SC35-0294
ADSM V3R1 Windows NT Quick Start	GC35-0295
ADSM V3R1 Using the UNIX Backup-Archive Client	SH26-4075
ADSM V3R1 Using the OS/2 Backup-Archive Client	SH26-4076
ADSM V3R1 Using the Novell NetWare Backup-Archive Client	SH26-4077
ADSM V3R1 Using the Microsoft Windows Backup-Archive Client	SH26-4078
ADSM V3R1 Installing the Clients	SH26-4080
ADSM V3R1 Using the Application Programming Interface	SH26-4081
ADSM V3R1 Trace Facility Guide	SH26-4082
ADSM V3R1 Client Reference Cards	SX26-6019
ADSM V3R1 Using the OS/2 Lotus Notes Backup Agent	SH26-4084
ADSMConnect Agent for Oracle7 Backup on AIX Installation and User’s Guide	SH26-4061
ADSMConnect Agent for Oracle Backup on Sun Solaris Installation and User’s Guide	SH26-4063
ADSMConnect Agent for Lotus Notes on Windows NT Installation and User’s Guide	SH26-4065
ADSMConnect Agent for Lotus Notes on AIX Installation and User’s Guide	SH26-4067
ADSMConnect Agent for Microsoft SQL Server Installation and User’s Guide	SH26-4069

B.4 ADSM Online Product Library

All of the ADSM publications are available in online readable format on the CD-ROM listed below.

CD-ROM Title	**Publication Number**
ADSM V3R1.0 MVS Online Product Library	SK3T-1396

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at `http://www.redbooks.ibm.com/`.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the World Wide Web**

  `http://w3.itso.ibm.com/`

- **PUBORDER** – to order hardcopies in the United States
- **Tools Disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLCAT REDPRINT
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type the following command:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

- **REDBOOKS Category on INEWS**
- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** – send orders to:

	IBMMAIL	**Internet**
In United States	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** – send orders to:

IBM Publications
Publications Customer Support
P.O. Box 29570
Raleigh, NC 27626-0570
USA

IBM Publications
144-4th Avenue, S.W.
Calgary, Alberta T2P 3N5
Canada

IBM Direct Services
Sortemosevej 21
DK-3450 Allerød
Denmark

- **Fax** – send orders to:

United States (toll free)	1-800-445-9269
Canada	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 408 256 5422 (Outside USA)** – ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	**Order Number**	**Quantity**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

□ Invoice to customer number

□ Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

Symbols

$ script variables 166
.adsmrc file 41
/usr/lib/x11/app-defaults directory, X Windows resource file 38

Numerics

2GB file sizes 87
3480/90 tape drives 99
3570/90 tape drives 99
4 mm tape drives 99
8 mm tape drives 99

A

ACLs, archiving 79
active restore session 68
administrative client
    BEGIN EVENTLOGGING command 304
    CANCEL RESTORE command 72
    client option sets 21, 42
    command line client 15
    compatibility with Version 2 15
    COPY PROFILE command 199
    COPY SCRIPT command 162
    DEFINE CLIENTACTION command 19, 155
    DEFINE CLIENTOPT command 150
    DEFINE CLOPTSET command 150
    DEFINE DEVCLASS command 248, 283
    DEFINE EVENTSERVER command 312
    DEFINE GRPMEMBER command 188
    DEFINE KEYRING command 121
    DEFINE PROFASSOCIATION command 197, 201
    DEFINE PROFILE command 199
    DEFINE SCRIPT command 161
    DEFINE SERVER command 180, 185, 188, 213, 246
    DEFINE SERVERGROUP command 188
    DEFINE SPACETRIGGER command 173
    DEFINE SUBSCRIPTION command 197, 203
    DELETE CLOPTSET command 151
    DELETE GRPMEMBER command 189
    DELETE PROFASSOCIATION command 202
    DELETE PROFILE command 200
    DELETE SCRIPT command 162
    DELETE SERVER command 189
    DELETE SUBSCRIBER command 202
    DELETE SUBSCRIPTION command 204
    DISABLE EVENTS command 303
    EDITOR option 82
    ENABLE EVENTS command 297, 302
    END EVENTLOGGING command 305
    GRANT AUTHORITY command 230, 232
    LABEL LIBVOL command 290
    LOCK PROFILE command 200
    MOVE DATA command 72
    national language support 89
    NOTIFY SUBSCRIBER command 202, 215
    ODBC driver 18
    PING SERVER command 186
    previous command recall 82
    QUERY ACTLOG command 299
    QUERY CLOPTSET command 151
    QUERY CONTENT command 96
    QUERY DRIVE command 285
    QUERY ENABLED command 303
    QUERY EVENTRULES command 304
    QUERY NODE command 153
    QUERY OCCUPANCY command 96
    QUERY PROFILE command 200, 204
    QUERY RESTORE command 68, 71
    QUERY SCRIPT command 162
    QUERY SERVER command 186, 247
    QUERY SERVERGROUP command 188
    QUERY SPACETRIGGER command 174
    QUERY SQLSESSION command 137
    QUERY STGPOOL command 96
    QUERY SUBSCRIBER command 202, 217
    QUERY SUBSCRIPTION command 204, 216
    QUERY SYSTEM command 147
    QUERY VOLUME command 96
    RECONCILE VOLUMES command 258
    REGISTER NODE command 151, 229, 244
    RENAME SCRIPT command 162
    RENAME STGPOOL command 209
    RUN command 162
    session numbers, restartable restores 72
    SET CONFIGMANAGER ON command 195, 197
    SET CONFIGREFRESH command 204
    SET CROSSDEFINE ON command 182
    SET INVALIDPWLIMIT command 158
    SET MINPWLENGTH command 159
    SET SERVERHLADDRESS command 183
    SET SERVERLLADDRESS command 183
    SET SERVERNAME command 181, 183
    SET SERVERPASSWORD command 181, 183
    SET SQLDATETIMEFORMAT command 136
    SET SQLDISPLAYMODE command 136
    SET SQLMATHMODE command 136
    SET WEBAUTHTIMEOUT command 119
    SETOPT command 145
    SQL interface
        see SELECT command
    UNLOCK PROFILE command 200
    UPDATE CLOPTSET command 151
    UPDATE NODE command 151
    UPDATE PROFILE command 199
    UPDATE SCRIPT command 162
    UPDATE SPACETRIGGER command 174
    Web administration
        see Web administrative interface
    Win32 GUI 15, 123
administrative command routing
    see enterprise administration
aggregate data transfer rate, calculation 37
aggregation, small client files 11, 93, 95

- allowreplace option, DEFINE SERVER command 213
- ANE prefix, client events 295
- archive
  - ACLs 79
  - adding files to package 76
  - creating new package 76
  - default description 75
  - deleting files from package 76
  - directories 79
  - DIRMC option, directories 80
  - packages 10, 75
  - retrieving files from package 76
  - trustee rights 79
  - using the GUI client 78

B

- backup-archive
  - GUI 7
- backup-archive client
  - 2GB file sizes 87
  - ACLs, archiving 79
  - active restore session 68
  - aggregate data transfer rate 37
  - aggregation, small files 11, 93
  - archive description 10, 75
  - archive packages 10, 75
  - calculated number of bytes, estimated 41
  - CANCEL RESTORE command 71
  - client option sets 21, 42, 144, 149
  - CLOPTSET option 151
  - collapsible directory tree, GUI 36
  - COMMRESTARTDURATION option 66, 103
  - COMMRESTARTINTERVAL option 66, 103
  - communication errors, recovery 66, 103
  - COMPRESSALWAYS option 107
  - confirms with server 104
  - conversion process, archive packages 75
  - directory archiving 79
  - directory selection, GUI 37
  - directory structure, restore options 10
  - DIRMC option, directory archiving 80
  - DIRSONLY option, archiving 79
  - DIRSONLY option, backup 47
  - dsm.opt, configuration 48
  - dsm.sys, configuration 48
  - duration, restartable restores 73, 103
  - EDITOR option 82
  - estimate function 41
  - event logging
    - see event management
  - EXCLUDE.DIR option 10, 48
  - EXCLUDE.FILE option 48
  - fault tolerance 10
  - file details, GUI 40
  - file errors, recovery 10, 65, 103
  - FILESONLY option, archiving 79
  - FILESONLY option, backup 47
  - find function, GUI 39
  - GUI 34
  - hard links, UNIX clients 86
  - historical data, estimate function 41
  - HTML help 88
  - INCLEXCL option 151
  - include/exclude options 48
  - incremental backup 9, 46, 106
  - incremental by date 59
  - LARGECOMMBUFFERS option 98
  - locale support 89
  - logical files, aggregation 93
  - Macintosh Service, Windows NT 86
  - memory constraints 11, 52
  - Motif support, GUI 34
  - national language support 58, 89
  - network data transfer rate 37
  - No Query Restore protocol 11, 52
  - online help 88
  - options editor 42
  - password length 159
  - PASSWORDDIR option 85
  - performance, small file aggregation 93
  - PITDATE restore option 58
  - PITTIME restore option 58
  - platforms supported 337, 338
  - point-in-time restore 9, 56
  - PRESERVEPATH option 50
  - previous command recall 82
  - QUERY RESTORE command 70
  - RESTART RESTORE command 70
  - restartable restore session 68
  - restartable restores 10
  - restore options 47
  - restore sessions, states 68
  - retention period, directory archives 80
  - REVOKEREMOTEACCESS client option
    - see Web client
  - selective backup, empty directories 9, 47
  - server 104
  - signon security 157
  - sort function, GUI 36
  - status bar, GUI 37
  - symbolic links, UNIX clients 86
  - tape mount prompt 62
  - TODATE restore option 56
  - tool bar, GUI 36
  - TOTIME restore option 56
  - transaction size 69
  - transfer time, estimated 41
  - trustee rights, archiving 79
  - TXNBYTELIMIT option 69, 93, 96
  - UNIX special files 86
  - USELARGEBUFFERS option 98
  - volume errors, recovery 10, 65, 103
  - Web interface
    - see Web client
  - X Windows resources 38
- banner frame, Web administrative interface 112
- BEGIN EVENTLOGGING command 304
- block sizes, tape 99

C

CANCEL RESTORE command 72
  administrative client 72
  backup-archive client 71
certificate request, SSL 121
certificates, SSL 120
certification authority, SSL 121
CERTUTIL utility, SSL 121
client acceptor, Web client 234
client access authority
  see Web client
client option sets 21, 42, 144, 149
client owner authority
  see Web client
CLOPTSET client option 151
collapsible directory tree, backup-archive GUI 36
command line client, Web administrative interface 111, 113, 114
command routing
  see enterprise administration
commands, administrator
  BEGIN EVENTLOGGING 304
  CANCEL RESTORE 72
  COPY PROFILE 199
  COPY SCRIPT 162
  DEFINE CLIENTACTION 19, 155
  DEFINE CLIENTOPT 150
  DEFINE CLOPTSET 150
  DEFINE DEVCLASS 248, 283
  DEFINE EVENTSERVER 312
  DEFINE GRPMEMBER 188
  DEFINE KEYRING 121
  DEFINE PROFASSOCIATION 197, 201
  DEFINE PROFILE 199
  DEFINE SCRIPT 161
  DEFINE SERVER 180, 185, 188, 213, 246
  DEFINE SERVERGROUP 188
  DEFINE SPACETRIGGER 173
  DEFINE SUBSCRIPTION 197, 203
  DELETE CLOPTSET 151
  DELETE GRPMEMBER 189
  DELETE PROFASSOCIATION 202
  DELETE PROFILE 200
  DELETE SCRIPT 162
  DELETE SERVER 189
  DELETE SUBSCRIBER 202
  DELETE SUBSCRIPTION 204
  DISABLE EVENTS 303
  ENABLE EVENTS 297, 302
  END EVENTLOGGING 305
  GRANT AUTHORITY 230, 232
  LABEL LIBVOL 290
  LOCK PROFILE 200
  MOVE DATA 72
  NOTIFY SUBSCRIBER 202, 215
  PING SERVER 186
  QUERY ACTLOG 299
  QUERY CLOPTSET 151
  QUERY CONTENT 96
  QUERY DRIVE 285
  QUERY ENABLED 303
  QUERY EVENTRULES 304
  QUERY NODE 153
  QUERY OCCUPANCY 96
  QUERY PROFILE 200, 204
  QUERY RESTORE 68, 71
  QUERY SCRIPT 162
  QUERY SERVER 186, 247
  QUERY SERVERGROUP 188
  QUERY SPACETRIGGER 174
  QUERY SQLSESSION 137
  QUERY STGPOOL 96
  QUERY SUBSCRIBER 202, 217
  QUERY SUBSCRIPTION 204, 216
  QUERY SYSTEM 144, 147
  QUERY VOLUME 96
  RECONCILE VOLUMES 258
  REGISTER NODE 151, 229, 244
  RENAME SCRIPT 162
  RENAME STGPOOL 209
  RUN 162
  SET CONFIGMANAGER ON 195, 197
  SET CONFIGREFRESH 204
  SET CROSSDEFINE ON 182
  SET INVALIDPWLIMIT 158
  SET MINPWLENGTH 159
  SET SERVERHLADDRESS 183
  SET SERVERLLADDRESS 183
  SET SERVERNAME 181, 183
  SET SERVERPASSWORD 181, 183
  SET SQLDATETIMEFORMAT 136
  SET SQLDISPLAYMODE 136
  SET SQLMATHMODE 136
  SET WEBAUTHTIMEOUT 119
  SETOPT 145
  UNLOCK PROFILE 200
  UPDATE CLOPTSET 151
  UPDATE NODE 151
  UPDATE PROFILE 199
  UPDATE SCRIPT 162
  UPDATE SPACETRIGGER 174
  WAIT=YES option 19, 175
commands, client
  CANCEL RESTORE 71
  DIRSONLY option 47
  FILESONLY option 47
  INCREMENTAL 46
  PITDATE restore option 58
  PITTIME restore option 58
  PRESERVEPATH restore option 50
  QUERY RESTORE 70
  RESTART RESTORE 70
  TODATE restore option 56
  TOTIME restore option 56
COMMETHOD HTTP server option 116
COMMETHOD HTTPS server option 121
COMMETHOD SNMP server option 331
COMMRESTARTDURATION client option 66, 103
COMMRESTARTINTERVAL client option 66, 103
communication errors, client 66, 103

communications, server-to-server
  cross definition, servers 180, 182
  definition, server groups 188
  functions 22, 177, 179
    see also enterprise administration
  server definitions 22, 180
  server groups 187
  server-to-server event logging
    see event management
  TCP/IP protocol 22, 180
  terminology 179
COMPRESSALWAYS client option 107
configuration manager
  see enterprise configuration
configuration profiles
  see enterprise configuration
configuration view, Web administrative interface 113
conversion process, archive packages 75
COPY PROFILE command 199
COPY SCRIPT command 162
copygroup parameter, point-in-time restore 59, 61
cross definition, servers 180, 182
CROSSDEFINE option, DEFINE SERVER command 185

D

DB2 integration with ADSM 2
default description, archive 75
default_profile, enterprise configuration 197
DEFINE CLIENTACTION command 19, 155
DEFINE CLIENTOPT command 150
DEFINE CLOPTSET command 150
DEFINE DEVCLASS command 248, 283
DEFINE EVENTSERVER command 312
DEFINE GRPMEMBER command 188
DEFINE KEYRING command 121
DEFINE PROFASSOCIATION command 197, 201
DEFINE PROFILE command 199
DEFINE SCRIPT command 161
DEFINE SERVER command 180, 185, 188, 213, 246
DEFINE SERVERGROUP command 188
DEFINE SPACETRIGGER command 173
DEFINE SUBSCRIPTION command 197, 203
DELETE CLOPTSET command 151
delete files, from archive package 76
DELETE GRPMEMBER command 189
DELETE PROFASSOCIATION command 202
DELETE PROFILE command 200
DELETE SCRIPT command 162
DELETE SERVER command 189
DELETE SUBSCRIBER command 202
DELETE SUBSCRIPTION command 204
DELGRACEPERIOD option, server-to-server 257
DES encryption
  see Web administrative interface
description, archive 10, 75
detail frame, Web administrative interface 113, 114
directory archiving 79
directory level incremental backup 9, 46
directory selection, backup-archive GUI 37
directory structure, restore options 10
DIRMC option, directory archiving 80
DIRSONLY option
  archive/retrieve operations 79
  backup/restore operations 47
DISABLE EVENTS command 303
Disaster Recovery Manager
  see DRM
disk I/O buffers 12
Distributed Protocol Interface
  see DPI
  see SNMP
DLT tape drives 99
DPI 30
drive mount limits 28
DRM 2
dsm.ini file 41
dsm.opt options file 42, 48
dsm.sys options file 42, 48
DSMAGENT, remote client agent 234
DSMCAD, client acceptor 234
DSMSERV.IDL job 117
dsmserv.opt, server options file 331
dsmsnmp process 326
DTF tape drives 99
duration, restartable restores 73

E

EDITOR client option 82
empty directories, selective backup 9, 47
ENABLE EVENTS command 297, 302
encrypted password, client 84
END EVENTLOGGING command 305
enterprise administration
  administrative command routing 177, 190
    administrative userid 191
    configuration 191
    syntax, commands 191
  enterprise configuration
    see enterprise configuration
  license 191, 195, 203
  server-to-server communications
    see communications, server-to-server
enterprise configuration
  commands 199
  configuration conflicts 205
  configuration manager 194, 195
  configuration profiles 194, 197, 198
  configuration refresh 204, 215
  default_profile 197
  license 203
  managed objects 194, 198
    administrative command schedules 198, 210
    administrator userids 198, 206
    client option sets 198, 212
    policy domains 198, 208
    server and server groups 213
    server scripts 198, 211
    servers and server groups 198
  managed servers 194, 203, 215
  overview 24

profile subscribers 194, 198, 203, 215
enterprise console
see Web administrative interface 110
estimate function, backup-archive client 41
event management
ANE prefix, client events 295
API messages 294
client event logging 292
client statistics 293
error messages, client 293
event receivers 300
eventserver receiver 312
file exit receiver 306
NetView receiver 317
NT eventlog receiver 310
overview 29
server-to-server event logging 179, 312
Tivoli event receiver 314
user exit receiver 308
event receivers 300
event viewer, Web administrative interface 111, 113, 115
eventserver event receiver 312
EXCLUDE.DIR option 10, 48
EXCLUDE.FILE option 48
EXIT statement, server scripts 167
expansion, database and recovery log 171

F
Fast Path, VTAM 13, 93
fault tolerant clients 10
file aggregation, client files 93
file details function, backup-archive GUI 40
file errors, recovery 10, 65, 103
file exit event receiver 306
FILESONLY option
archive/retrieve operations 79
backup/restore operations 47
find function, backup-archive GUI 39
FREQUENCY copygroup parameter 60

G
GOTO statement, server scripts 167
GRANT AUTHORITY command 230, 232
GUI, backup-archive client 34

H
hard links, UNIX clients 86
heartbeat monitor, SNMP 30
HLADDRESS option, DEFINE SERVER command 185
HPDT 13, 93
HTML help, clients 88
HTTP COMMMETHOD 116
HTTP communications protocol 116, 122
HTTP daemon 116
HTTPPORTserver option 116
HTTPS communications protocol 121, 122
HTTPSPORT server option 121

I
I/O operation buffers 12
IF clause, server scripts 167
INCLEXCL client option 151
include/exclude client options 48
incremental backup granularity 106
incremental by date backup option 59
interface definition, Web administrative interface 117
Internet Explorer, supported levels 110, 227

J
Java applet, command line client 114
Java applet, event viewer 115
Java requirements
Web administrative interface 110
Web client 227

K
keyring file, SSL 120

L
LABEL LIBVOL command 290
LABEL statement, server scripts 167
LARGECOMMBUFFERS client option 98
last command recall 82
LLADDRESS option, DEFINE SERVER command 185
locale support 89
LOCK PROFILE command 200
logical file, file aggregation 95
logical files, server aggregation 93
logical occupancy, file aggregation 94, 95

M
Macintosh Service, Windows NT 86
managed objects, enterprise configuration
see enterprise administration
managed servers, enterprise configuration
see enterprise administration
Management Information Base
see SNMP, MIB
MAXCAPACITY, device class option 248
memory constraints, client 11, 52
migration considerations
archive packages 75
LARGECOMMBUFFERS client option 98
TXNBYTELIMIT client option 96
TXNGROUPMAX server option 96
USELARGEBUFFERS server option 99
USELARGETAPEBLOCK server option 99
MKKFE utility, SSL certificate creation 120
Motif support, backup-archive GUI 34
mount limits, drive 28, 282–286
MOVE DATA command 72

N
national language support 58, 89
Netscape, supported levels 110, 227

NetView event receiver 317
network data transfer rate, calculation 37
network view, Web administrative interface 112
No Query Restore protocol 11, 52, 102
NOTIFY SUBSCRIBER command 202, 215
NT eventlog event receiver 310

O
object view, Web administrative interface 113
ODBC
  API 138
  applications 18, 140
  configuration 140
  data sources 140
  driver 18, 138
  implementation 139
online help, clients 88
Open Database Connectivity
  see ODBC
operation view, Web administrative interface 112
options editor, backup-archive client 42
overflow storage pools 28, 270–278

P
packages, archive 10
password length, clients 159
PASSWORDACCESS GENERATE client option 84
PASSWORDDIR client option 85
passwords, Web client 234
PDF documentation 89
performance, small file aggregation 93
physical file, file aggregation 95
physical occupancy, file aggregation 95
PING SERVER command 186
PITDATE restore option 58
PITTIME restore option 58
point-in-time restore
  national language support 58
  No Query Restore protocol 59
  policy definitions 59, 61
  requirements 59
  restore options 58
policy definitions, point-in-time restore 59, 61
PRESERVEPATH restore option 50
previous command recall 82
profile subscribers
  see enterprise configuration

Q
QIC tape drives 99
QUERY ACTLOG command 299
QUERY CLOPTSET command 151
QUERY CONTENT command 96
QUERY DRIVE command 285
QUERY ENABLED command 303
QUERY EVENTRULES command 304
QUERY NODE command 153
QUERY OCCUPANCY command 96
QUERY PROFILE command 200, 204
QUERY RESTORE command
  administrative client 68, 71
  backup-archive client 70
QUERY SCRIPT command 162
QUERY SERVER command 186, 247
QUERY SERVERGROUP command 188
QUERY SPACETRIGGER command 174
QUERY SQLSESSION command 137
QUERY STGPOOL command 96
QUERY SUBSCRIBER command 202, 217
QUERY SUBSCRIPTION command 204, 216
QUERY SYSTEM command 144, 147
QUERY VOLUME command 96

R
RECLAIM parameter, storage pool 94
RECLAIMSTGPOOL parameter, storage pool 287
reclamation, virtual volumes 257
RECONCILE VOLUMES command 258
REGISTER NODE command 151, 229, 244
remote client agent, Web client 234
RENAME SCRIPT command 162
RENAME STGPOOL command 209
RESTART RESTORE command 70
restartable restores
  client filespace considerations 72
  commands 68, 70, 71, 72
  RESTOREINTERVAL option 73, 103
  server considerations 72
  sessions 68
restore options, command line client 47
RESTOREINTERVAL server option 73, 103
retention period, directory archives 80
RETEXTRA copygroup parameter 59, 61
RETONLY copygroup parameter 59, 61
retrieve files, from archive package 76
return codes, server scripts 166
REVOKEREMOTEACCESS client option 235
RUN command 162

S
SAP integration with ADSM 2
scripts, server 160
Secure Sockets Layer
  see SSL
security, client passwords 157
SELECT command
  AVG clause 130
  COUNT clause 130
  description 18
  FROM clause 130
  GROUP BY clause 132
  HAVING clause 132
  MAX clause 130
  ODBC driver 127
  ORDER BY clause 132
  SUM clause 130
  syntax 127, 128, 130, 131, 132, 134

  - SYSCAT.COLUMNS 128
  - SYSCAT.ENUM 128
  - SYSCAT.TABLES 128
  - WHERE clause 131
- selective backup of empty directories 47
- self-signed certificates, SSL 121
- server
  - 3480/90 tape drives 99
  - 3570/90 tape drives 99
  - 4 mm tape drives 99
  - 8 mm tape drives 99
  - active restore session 68
  - agent, SNMP 30
  - aggregate file 95
  - aggregation, small files 11, 93
  - client option sets 21, 42
  - COMMETHOD HTTP option 116
  - COMMETHOD HTTPS option 121
  - COMMETHOD SNMP option 331
  - communication buffers 12
  - confirms with client 104
  - conversion process, archive packages 75
  - database expansion 171
  - database threshold 171
  - disk I/O buffers 12
  - DLT tape drives 99
  - drive mount limits 28
  - dsmserv process 326
  - dsmserv.opt options file 331
  - dsmsnmp process 326
  - DTF tape drives 99
  - duration, restartable restores 73, 103
  - event logging
    - see event management
  - EXIT statement, scripts 167
  - FREQUENCY copygroup parameter 60
  - GOTO statement, scripts 167
  - heartbeat monitoring, SNMP 30
  - HPDT support 13, 93
  - HTTPPORT option 116
  - HTTPSPORT option 121
  - I/O operation buffers 12
  - IF clause, scripts 167
  - LABEL statement, scripts 167
  - logical files, aggregation 93, 95
  - logical occupancy, storage pool 94, 95
  - mount limits, drive 282–286
  - national language support 89
  - No Query Restore protocol 11, 52
  - overflow storage pools 28, 270–278
  - performance, small file aggregation 93
  - physical file 95
  - physical occupancy 95
  - platforms supported 336
  - point-in-time restore 9, 56
  - processes, affected by restartable restore 72
  - QIC tape drives 99
  - RECLAIM storage pool parameter 94
  - recovery log expansion 171
  - recovery log threshold 171
  - restartable restores 68
  - restore sessions, states 68
  - RESTOREINTERVAL option 73, 103
  - RETEXTRA copygroup parameter 59, 61
  - RETONLY copygroup parameter 59, 61
  - sample scripts 169
  - script return code 166
  - script variables 166
  - scripts 160
  - server-to-server communications
    - see communications, server-to-server
  - single drive reclamation 287
  - SNMP monitoring
    - see SNMP
  - SNMPSUBAGENT option 331
  - subagent, SNMP 30
  - tape block sizes 99
  - tape management 241
  - transaction size 69
  - traps, SNMP 30
  - triggers, database and recovery log utilization 172
  - TXNGROUPMAX option 69, 93, 96
  - USELARGEBUFFERS option 98
  - USELARGETAPEBLOCK option 99
  - VERDELETED copygroup parameter 59, 61
  - VEREXIST copygroup parameter 59, 61
  - virtual volumes
    - see virtual volumes, server-to-server
  - Web administration
    - see Web administrative interface
- server groups 188
  - see also communications, server-to-server 187
- SERVERPASSWORD option, DEFINE SERVER command 185
- server-to-server communications
  - see communications, server-to-server
- session numbers, restartable restores 72
- SET CONFIGMANAGER ON command 195, 197
- SET CONFIGREFRESH command 204
- SET CROSSDEFINE ON command 182
- SET INVALIDPWLIMIT command 158
- SET MINPWLENGTH command 159
- SET SERVERHLADDRESS command 183
- SET SERVERLLADDRESS command 183
- SET SERVERNAME command 181, 183
- SET SERVERPASSWORD command 181, 183
- SET SQLDATETIMEFORMAT command 136
- SET SQLDISPLAYMODE command 136
- SET SQLMATHMODE command 136
- SET WEBAUTHTIMEOUT command 119
- SETOPT command 145
- single administrative login
  - see Web administrative interface
  - see Web client
- single drive reclamation 287
- small file aggregation 11, 93
- SNMP
  - agent 30
  - COMMETHOD SNMP server option 331
  - DPI 30

dsmserv.opt options file 331
dsmsnmp process 326
MIB 329
monitoring 30
server monitoring 329
snmpd process 326
SNMPSUBAGENT server option 331
subagent 30
traps 30
sort function, backup-archive GUI 36
SQL interface 18
SSL
browser requirements 120
certificate request 121
certificates 120
certification authority 121
CERTUTIL utility 121
HTTP protocol 122
HTTPS protocol 121, 122
HTTPSPORT server option 121
keyring file 120
MKKFE utility 120
self-signed certificates 121
session encryption 120
Web administrative interface 110
status bar, backup-archive GUI 37
symbolic links, UNIX clients 86
syntax, administrative command routing 191

T
tape block sizes 99
tape management 241
tape mount prompt 62
Tivoli event adapter 314
TODATE restore option 56
tool bar, backup-archive GUI 36
TOTIME restore option 56
transaction size, client and server 69, 93
transfer rates, backup-archive client 37
tree frame, Web administrative interface 112, 114
triggers, database and recovery log utilization 172
trustee rights, archiving 79
TXNBYTELIMIT client option 69, 93, 96
TXNGROUPMAX server option 69, 93, 96
type=server parameter, REGISTER NODE command 244

U
UNIX special files 86
UNLOCK PROFILE command 200
UPDATE CLOPTSET command 151
UPDATE NODE command 151
UPDATE PROFILE command 199
UPDATE SCRIPT command 162
UPDATE SPACETRIGGER command 174
URL parameter, DEFINE SERVER command 223
URL, clients 220
URL, HTTP prefix 122
URL, HTTPS prefix 122
URL, servers 220
USELARGEBUFFERS client option 98
USELARGETAPEBLOCK server option 99
user exit event receiver 308
userid parameter, REGISTER NODE command 230
utilization trigger, database and recovery log 172

V
variable substitution, server scripts 166
VERDELETED copygroup parameter 59, 61
VEREXIST copygroup parameter 59, 61
virtual volumes, server-to-server
deleting virtual volumes 257
DELGRACEPERIOD option 246, 257
device class 243, 248
DRM integration 261–268
expiration 257
license 243
MAXCAPACITY, device class option 248
nodename, source server 246
overview 26
reclamation 257
reclamation, virtual volumes 257
type=server parameter, REGISTER NODE command 244
volume names 254
volume errors, recovery 10, 65, 103
VTAM High Performance Data Transfer
see HPDT

W
WAIT=YES option, server commands 19, 175
Web administrative interface
banner frame 112
browser requirements 110
certificate request 121
certification authority 121
command line 111, 113, 114
configuration view 113
DES encryption 223, 224
detail frame 113, 114
DSMSERV.IDL job 117
enterprise console 110
event viewer 111, 113, 115
HTTP protocol 116, 122
HTTPS protocol 122
hyperlinks 220
interface definition job 117
Internet Explorer 110
Java applet 114, 115
Netscape 110
network view 112
object view 113
operation view 112
security 118
self-signed certificates, SSL 121
server scripts 164
session logging 119
single administrative login 220
SSL 110, 120

- SSL certificates 120
- tree frame 112, 114
- URL 220, 224
- user interface 114

Web client
- administrative authorities 228–233, 236–240
- authentication 17, 237–240
- client acceptor 16, 234
- client access authority 228–233, 236–238
- client functions 16, 228
- client owner authority 228–231, 236–238
- configuration 234
- DES encryption 236, 237
- enterprise console 110, 236, 237
- hyperlinks 220, 235, 236
- Java applet 16, 227
- PASSWORDACCESS client option 229, 234
- passwords 234
- remote client agent 16, 234
- REVOKEREMOTEACCESS client option 235, 238
- supported clients 227
- URL 220, 235, 236, 239
- userid 228–233
- Version 3.1.1 servers 240

Webshell client
- see Web client

Win32 administrative GUI 15, 123

Windows NT Macintosh Services 86

X

X Windows resources 38

XPG4 support 89

ITSO Redbook Evaluation

ADSM Version 3 Technical Guide
SG24-2236-01

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ Customer _ Business Partner _ Solution Developer _ IBM employee
_ None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction __________

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

ADSM Version 3 Technical Guide **SG24-2236-01**

SG24-2236-01
Printed in the U.S.A.