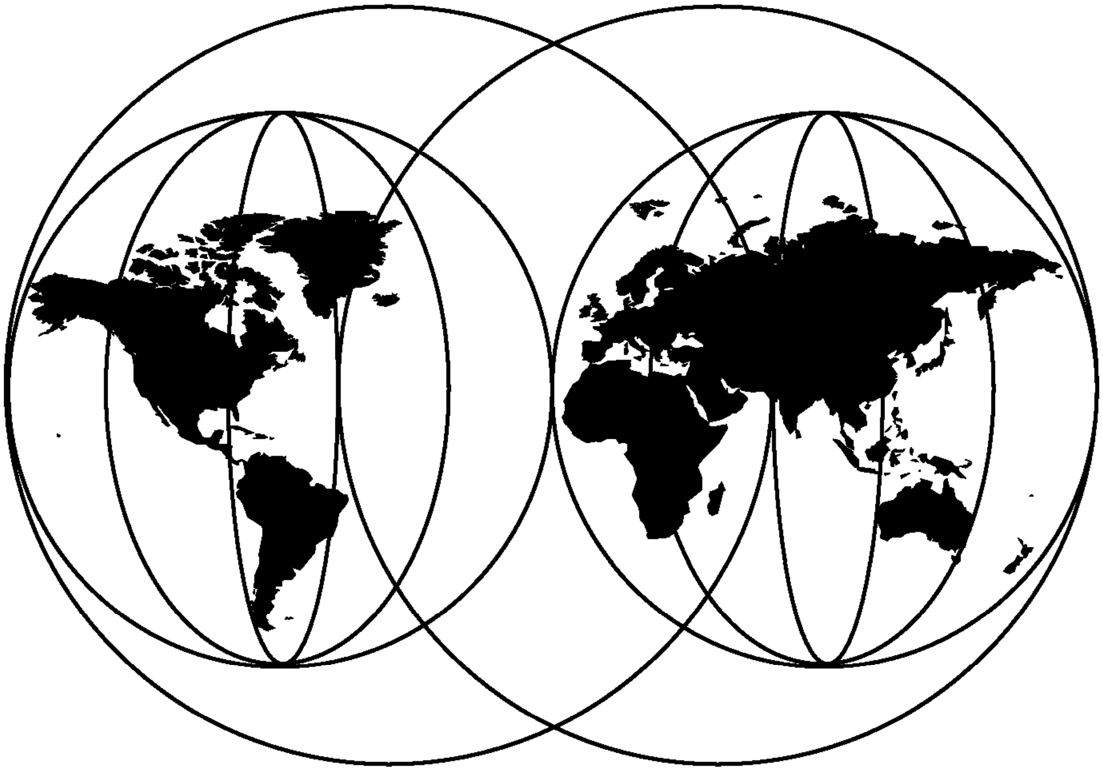
Image and Workflow Library: SmartGuide to EDMSuite System Managed Storage

Mike Ebbers   Juan F. Abal   Eric Cohen   Shantaveerayya K K



International Technical Support Organization

http://www.redbooks.ibm.com

This book was printed at 240 dpi (dots per inch). The final production redbook with the RED cover will be printed at 1200 dpi and will provide superior graphics resolution. Please see "How to Get ITSO Redbooks" at the back of this book for ordering instructions.

SG24-5159-00

International Technical Support Organization

SG24-5159-00

Image and Workflow Library: SmartGuide to EDMSuite System Managed Storage

June 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, "Special Notices" on page 203.

First Edition (June 1998)

This edition applies to Version 2, Release 3, Modification 1 of IBM ImagePlus VisualInfo (04L0256) for use with the Windows NT Version 4 and AIX Version 4.2.1 operating systems.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

Contents

Preface

One of the more complex portions of an ImagePlus VisualInfo system is System Managed Storage (SMS). This redbook examines SMS on two platforms: IBM AIX and Windows NT. It provides guidance for installing and configuring SMS and ADSM for VisualInfo V2.3.1 on these platforms, and gives you hints and tips on how to avoid or solve problems.

The tailoring and operation of ADSM is an important part of this redbook, since it is a mandatory component when using the optical library for VisualInfo storage. So we cover ADSM in depth.

This redbook was written for system programmers, consultants and other technical specialists who work with VisualInfo and Digital Library. We assume that the reader has a basic knowledge of each of the products.

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 Raleigh and San Jose Centers.

Mike Ebbers was the project leader for this redbook. He is a Senior International Support Representative at the International Technical Support Organization, Poughkeepsie Center. He has worked at IBM for 24 years. He produces redbooks on workflow and image products. Mike previously developed and taught training courses on ImagePlus.

Juan F. Abal is an I/T Architect in Spain. He has 12 years of experience in image products. He holds a degree in Computer Sciences from the Universidad Complutense de Madrid. His areas of expertise include e-Document Management and network computing.

Eric Cohen is a Technical Service Specialist in France. He has 13 years of experience in the computer field. He holds a degree in Electronic Engineering from the Institut National des Sciences Appliquees in Lyon. For the past year he has worked on image and workflow systems.

Shantaveerayya K K is a Technical Executive in Product Support Services in India. He has one year of experience in the workflow and imaging fields. He holds a degree in Electrical Engineering from Regional Engineering College Bhopal (India). His areas of expertise include workflow, imaging, and speech recognition systems.

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

Stuart Byrom
Rick Salmon
Jackie Williams
IBM VisualInfo Testing Group, Raleigh Lab

Jim Lines
Leonora Wang
IBM Digital Library Testing Group, Santa Teresa Lab

Gerald Kozina
Phillip Sanchez
IBM Digital Library Development Group, Santa Teresa Lab

Tim Mortimer
IBM 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 237 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. Creating a Solution

The paperless society is becoming a reality, since many companies are storing their information in other media, such as images, audio, video, graphics and spread sheets, as well as machine readable text. IBM offers storage and retrieval solutions for these media, such as ImagePlus VisualInfo and Digital Library. Organizations large and small are using these products around the world. They allow users to capture, store, and retrieve documents online and provide document, folder, and workflow management capabilities. They also provide extensive data integrity and security.

With the ImagePlus systems, you can develop a customized document management solution that includes library and information processing capabilities for multiple media types. You can create image, workflow, facsimile (using IBM FaxRouter/2) and other applications to automate and gain control of the information your enterprise processes each day. You can increase productivity, lower storage costs, and improve customer service.

In this redbook, we cover the System Managed Storage (SMS) component of VisualInfo. Since SMS for Digital Library is functionally the same, the examples and references for VisualInfo can often apply to Digital Library.

1.1 Understanding the VisualInfo Solution

VisualInfo is a client/server solution. It has three main components:

- Library (or index) server
- Object server
- Client application

Each of the components can operate on any supported platform without operationally affecting any of the other components. The family of platforms is designed to grow from OS/2 and Windows through AIX and OS/400 to MVS/ESA.

1.1.1 System Overview

The VisualInfo system provides enterprise-wide access to document processing and library management. The system consists of one or more clients connected to one library server and one or more object servers. Here is a list of supported platforms:

- One or more Windows and OS/2-based client workstations (any mix)

- One OS/2, Windows NT, AIX, OS/400 or MVS-based library server

- One or more OS/2, NT, AIX, OS/400 and MVS-based object servers (any mix)

- One or more NT or OS/2 workstations to run the System Administration program

  Note: If you are mixing different versions of VisualInfo in your system, then certain server combinations may require that TCP/IP be used instead of SNA. Check the announcement materials or consult your IBM representative.

Figure 1 on page 2 shows the client/server relationship in the VisualInfo system. Departments located anywhere across an enterprise can access their own documents as well as enterprise documents. The system also provides for a distributed placement of documents and other data on a variety of LAN-based and host-based servers, while maintaining seamless user access.

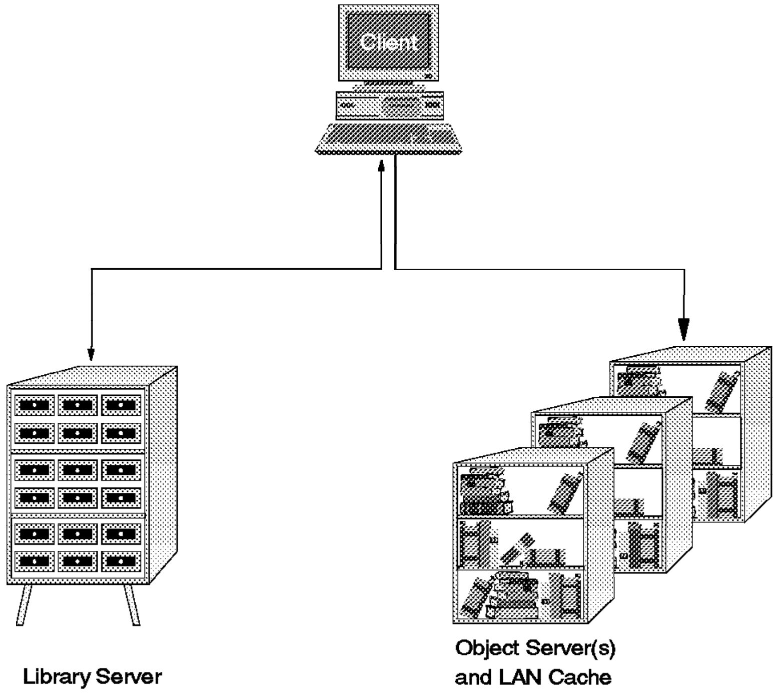


Figure 1. The Client/Server Relationship in the VisualInfo System

Typical end-user functions in a VisualInfo solution include:

- Storing a document
- Indexing a document
- Displaying a document
- Requesting a document from a workbasket

All of these activities use functions provided in the client and library server. That is, most VisualInfo functions requested by a user at the client workstation result in activity at the library server. The library server stores the indexing and retrieval information that you assign to each of your documents. The object servers are the actual repositories for your stored documents in the system.

Usually, the message flow from a client to the servers is:

1. The end user requests function on a client workstation.
2. The VisualInfo client does some processing and sends a message to the library server.
3. The library server does some processing and sends a message to the object server.
4. If necessary, an object moves from a client to an object server or from an object server to a client.
5. The object server sends a confirmation to the library server.
6. The library server sends a confirmation to the client, completing the transaction.

1.1.2 Understanding VisualInfo Communication Flow

The communications flow among client workstations, its assigned library server, and the available object servers follow a sequence that you can see in Figure 2.

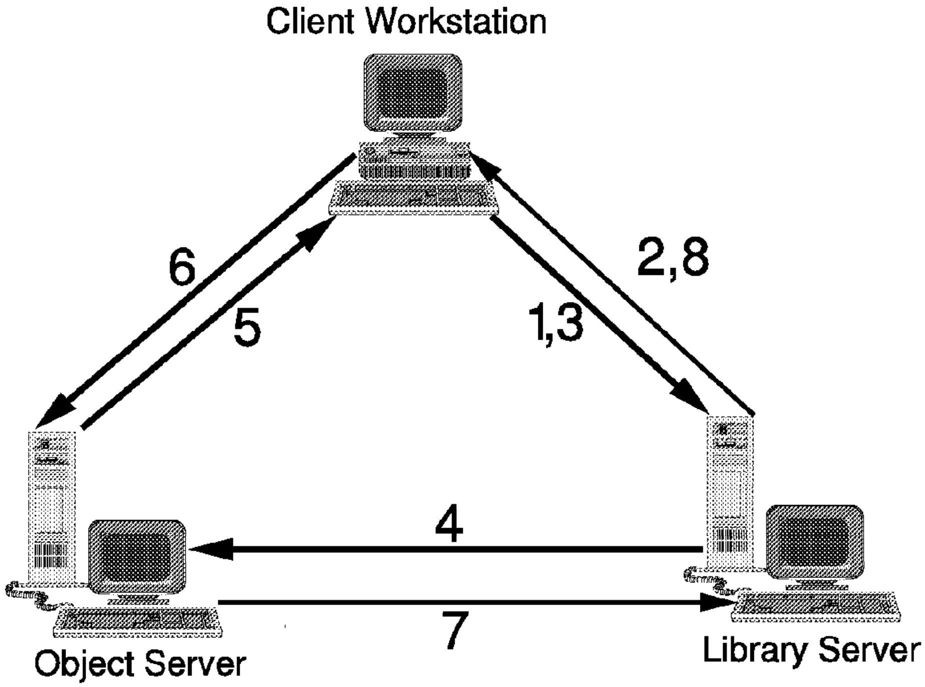


Figure 2. VisualInfo Communications Flow

Figure 2 shows a complete VisualInfo flow. Follow the sequence:

1. The client workstation makes an object (document) request to its assigned library server.

   Note: The Windows client does not support local cache. Jump to step 4.

2. If the client is on OS/2, the library server receives the request and sends back the object ID to the workstation client. This process is done because the client workstation checks to see if it has the object stored in its WORKSTATION LOCAL CACHE and if it is the latest update. If the object is found in the local cache and it is the latest update, the process finishes here.
3. If the object is not found in the local cache, the client workstation sends a notification to the library server saying that it does not have the object.
4. The library server sends an object request to the appropriate object server that owns the object. The client workstation ID that is requesting the object is also sent.
5. The object server sends the object to the client workstation.
6. The client workstation notifies the object server that the object was received successfully.

7. The object server notifies the library server that the operation was completed successfully.

8. The library server finishes the object request process. This process uses the DB2 two-phase commit feature.

1.1.3 Interaction of VisualInfo Components

1.1.3.1 Role of DB2

DB2 is used by VisualInfo to maintain all of the databases that control the objects that are stored in the system. Through DB2 databases and tables, VisualInfo knows, for example, where the objects are stored and which keys retrieve them.

DB2 performance is very important for VisualInfo overall performance, so database administration is a very important point to consider when you are planning your solution.

1.1.3.2 Role of TCP/IP or SNA

All communication among VisualInfo components (when they do not reside in the same workstation) is managed by TCP/IP or SNA (OS/2 only).

When a client machine performs an object request, TCP/IP (or SNA) is responsible for transporting this request to the library server workstation. They are also responsible for transporting the object from the object server workstation to the client machine.

1.1.3.3 Role of C++

VisualAge C++ (IBM OS/2) or Microsoft Visual C++ (Windows NT) provides the runtime libraries and the compiler that VisualInfo needs to build the DLLs for each index class that is created by the system.

If you have a development environment or if you want to create applications to work with VisualInfo, there are many VisualAge tools and samples that you can install to optimize your application programming.

1.1.4 A Closer Look at the VisualInfo System

To better understand the VisualInfo (or Digital Library) system, let us take a closer look at its components and the ways you can work with them.

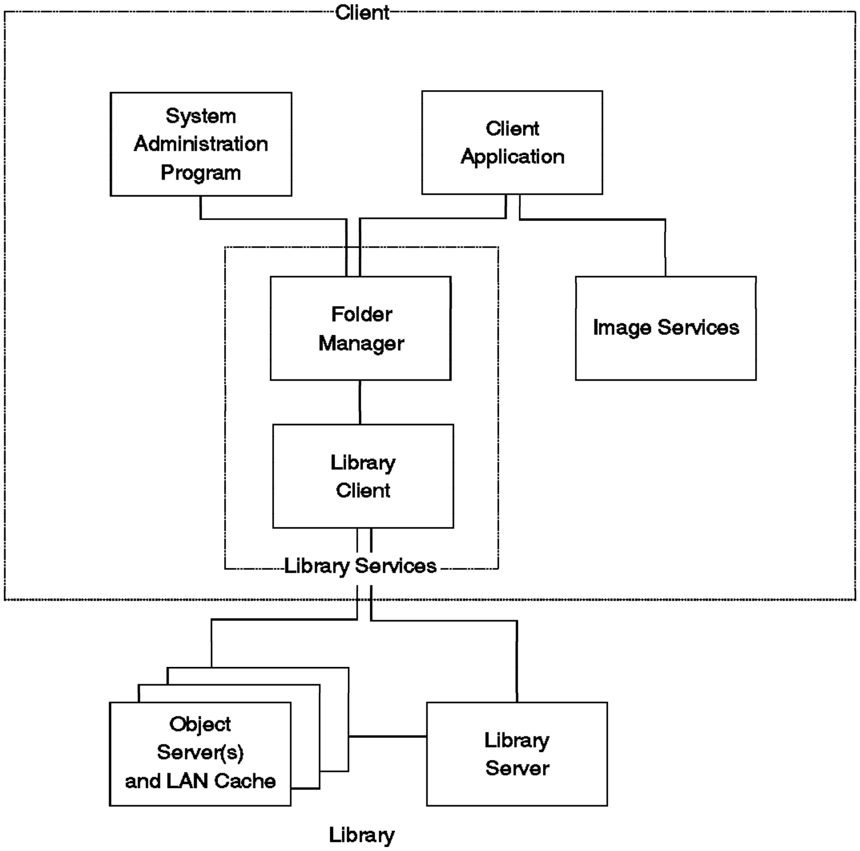


Figure 3. The Main Components in a VisualInfo Implementation

Figure 3 shows the relationship between the components of the system. An OS/2 example is shown, and other platforms are similar.

- Client components
  - Client Application (customizable)
  - Image Services
  - Folder Manager
  - Library Client
  - System Administration program
- Library Server
- Object Server

1.1.4.1 Client Features

The VisualInfo system includes a choice of configurable client applications that provide a complete end-user interface for your document management system.

The system also includes a wide variety of application programming interfaces (APIs) that let you use your business applications with the VisualInfo system and the available fax processing, workflow, and document capture complementary offerings.

User exits provide points where you can create application-specific processing routines to customize your VisualInfo system.

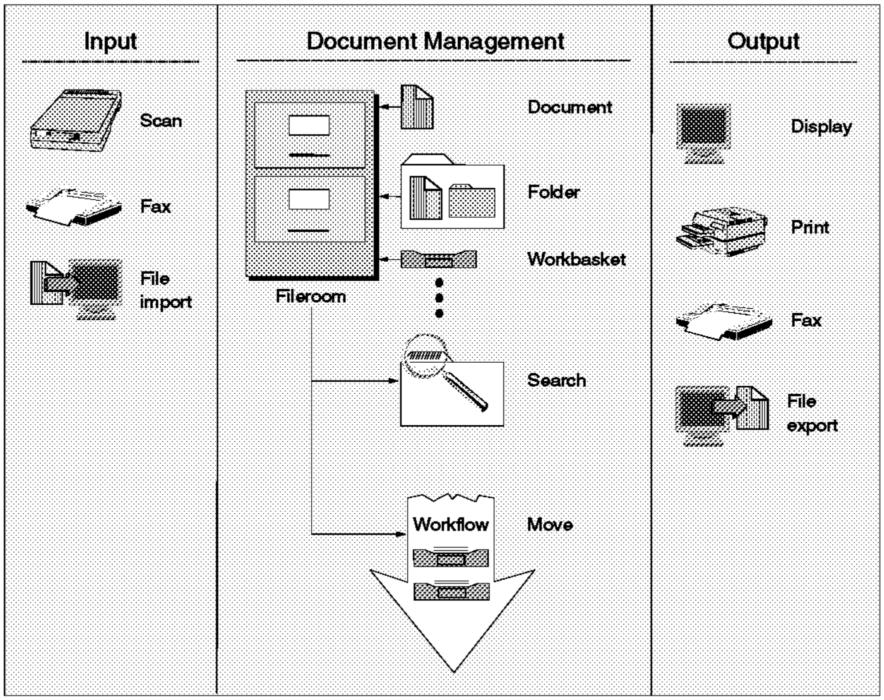


Figure 4. VisualInfo Client Implementation

Client Applications: The Windows and OS/2 client applications provide a complete imaging and document processing system that you can customize to meet the needs of your enterprise. The client applications interact with all of the VisualInfo components through an end-user interface. The client applications do not require additional programming; they integrate the features of the VisualInfo system, including the folder manager services, into one application.

The client applications let you input documents by scanning, importing, or fax input (through FaxRouter/2). Then you can manage the documents and the folders containing the documents in an online file room. You can search for documents and folders and move them through a predetermined workflow. You can also output documents by displaying, printing, or exporting files.

The client applications provide APIs to let you integrate folder, workflow, and document management with your existing information systems. You can easily integrate your custom software and other applications. Figure 4 shows a typical VisualInfo client implementation. You can use the client applications as an interface to all of your document management needs, customize the client applications, or use the Folder Manager, Library Client, and Image Services APIs to interface with your own applications.

Image Services: Image Services provides the capture (scanning support), display, and print services available through the client applications. It is a component of the OS/2 platform, but not the Windows NT platform. You can create your own customized application to capture, print, and display documents using the Image Services APIs.

Folder Manager: The folder manager provides a document management programming interface that uses the library client to access information stored on the library server and object servers.

The folder manager provides a data model for managing online documents and folders as you do with paper documents. It submits API requests to the library client to perform library functions for your applications. The library client then interacts with the library server and the object server independent of where the servers are. Because of this, you can expand your VisualInfo system without having to change your applications.

Your system can access these folder manager services through a client application. You can also interface directly with the VisualInfo system through the folder manager APIs to create a customized document management, workflow, or other business application.

Depending on your level of access to documents, you can perform the following operations using the folder manager.

- Store a document
- Index a document or folder
- Process a document or folder
- Retrieve a document or folder

Library Client: As an alternative to using the folder manager, you can use the library client, a lower-level interface that lets you create your own data model. The library client increases your flexibility in creating your own applications. It lets you interface directly with the library server and object server in your VisualInfo system through APIs. You can use the library API calls directly if you decide not to use the folder manager, or you can mix the two API sets in a customized application.

Compared to folder manager APIs, the library client APIs provide access to a more detailed level of the underlying product structure.

Figure 5 on page 8 shows how the client application and the VisualInfo APIs logically fit together.

You can also use library client APIs for services not available through the folder manager to take advantage of different performance characteristics.

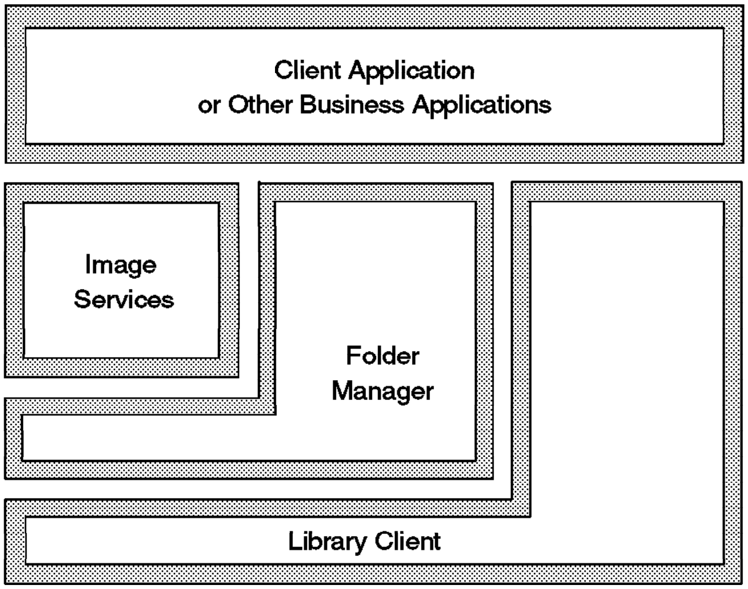


Figure 5. VisualInfo Layers

System Administration Program: The system administration program is a powerful end-user interface that lets you configure your VisualInfo system. It lets you manage all of your system components, no matter which platform they run on. The system administration program can reside on any workstation in your VisualInfo network.

You can use this program to configure resources such as workbaskets and workflow. The system administration program lets you control the level of access that users have to the VisualInfo system and customize the document management services to meet the specific needs of your enterprise.

1.1.4.2 Library Server Features

The library server uses relational database technology to maintain the library contents, and provides data integrity by performing the following functions:

- Managing library data
- Maintaining index information
- Controlling access to documents stored in the object servers

For instance, the library server routes requests from the library client for the following:

- To store, retrieve, and update documents stored in object servers
- To update and query the indexes and descriptive information stored in the library catalog

The Client Application can log onto only one library server at a time. However, you can develop applications to reference multiple library servers using the folder manager or the library client interface.

1.1.4.3 Object Server Features

Each object server is associated with one library server and maintains documents stored in the library. The object server receives requests from the library server and communicates with the library client to complete requests from your application.

Some object servers support attachment of both direct access storage devices (DASD) and a range of optical storage devices. They also provide system-managed storage to automatically move documents from one storage medium to another, such as from DASD to optical disk.

Object servers can be installed on any of the platforms (OS/2, NT, AIX, OS/400 and MVS) for maximum configuration flexibility and performance.

1.1.5 Growth Potential

VisualInfo is designed to provide a system that meets your current needs and can grow from there. You can start with a small LAN-based system and add AIX, AS/400 and MVS components as needed to grow vertically. You can also add additional servers, workstations, and LANs to grow outward.

1.1.5.1 Scalability

The scalability of the VisualInfo product provides a solution that covers stand-alone and local area network (LAN) systems, up to an enterprise environment where multiple departments of an enterprise, located in one or several locations, access local data as well as enterprise data.

Hence, user growth is simplified through the ability to support:

- Separate object and library servers
- Multiple object servers on the LAN
- MVS/ESA-based object and library servers

For example, a configuration can grow through the following stages:

1. Initial installation of a *stand-alone configuration* for demonstration or application development purposes
2. A *combined object server and library server* on a LAN with several client machines
3. *Separate object and library servers* on the LAN
4. A single library server and *multiple object servers*
5. Migration of the *library server to MVS/ESA* while continuing with LAN-based object servers for local storage of objects
6. Addition of a central archive *object server on MVS/ESA*

1.1.6 Selecting a Platform

The VisualInfo system is an open, scalable, production image solution that capitalizes on the benefits of a client/server architecture. The scalability of the VisualInfo solution comes in two forms:

- The ability to choose hardware platforms ranging from Intel-based machines to System/390 servers

- The ability to add object servers as your data volume or data distribution grows

When selecting platforms (OS/2, NT, AIX, OS/400 or MVS), the user should bear in mind the reliability, security, and control that a given platform can provide.

A customer may decide to install VisualInfo servers in MVS/ESA to achieve:

- The capability to support large applications (hundreds of users, millions of documents)
- Centrally managed, tight control
- A high degree of security
- A high degree of recoverability

If a customer decides to install VisualInfo servers on an OS/2 or NT platform, this allows:

- Easy installation, configuration, and management
- Fast deployment
- Scalability (start small)
- A high degree of control by end users
- The capability to support small and medium applications

A customer may choose the AIX platform for servers, and achieve performance results closer to MVS than to OS/2, but without the security, recovery and central management that MVS offers.

Your library server can be on any platform, independent of other server or client platforms. You can mix object servers in OS/2 or NT, AIX, OS/400 and MVS platforms. This can be a good way to improve the retrieval performance of remote documents that are temporarily stored in a local object server.

Note: When mixing versions older than V2.2, check product documentation to see which releases of clients and servers can work together.

You can also have a local library server in OS/2 working with both a local LAN cache object server (OS/2) and a remote object server (NT, AIX or MVS) together.

1.1.7 Additional Components of Digital Library

In addition to the previous description of VisualInfo, which applies to Digital Library as well, there are two other components that come with Digital Library.

1.1.7.1 Text Search Server

The Text Search Server performs indexing and retrieving documents based on words and topics within the text. This is a software prerequisite for the Application Development Toolkit for AIX.

1.1.7.2 Internet Connection

The Internet Connection performs accessing of data stored in the object server through a Web interface.

1.1.7.3 Interface with QBIC

If you have the Query by Image Content (QBIC) product, the Image Search Server provides the QBIC capability which allows users to search images by specifying an example source image or pre-extracted image content features. Query by example and the similarity search are two key characteristics of QBIC, where the search results are ranked by the similarity of the target images to example source image. The supported image features include:

- Average color—average color in the whole image
- Histogram color—distribution of colors in an image
- Positional color—average color in a specific area of an image
- Texture—coarseness, contrast and directionality of an image

1.2 Introduction to ADSM

This section provides a brief overview of ADSM, the market′s best storage manager in a distributed environment. We look at the many platforms ADSM supports, the main components of ADSM, and descriptions of key functions such as scheduling and policy management. We conclude with a summary of ADSM advantages.

1.2.1 Use of ADSM by Each Platform

On AIX and NT, VisualInfo delegates the management of object storage volumes to ADSM. VisualInfo on OS/2 does not use ADSM, but manages the objects itself, using System Managed Storage. On MVS, VisualInfo relies on OAM for object management (this is not covered in this redbook).

1.2.2 What Is the ADSTAR Distributed Storage Management System?

ADSTAR Distributed Storage Management system (ADSM) is a client/server program product. It provides IBM′s solution to enterprise-wide distributed storage management, highly automated, centrally scheduled, network-based backup and archive functions for workstations and LAN file servers. ADSM supports a wide variety of IBM and non-IBM clients and servers, as shown in Figure 6 on page 12, and addresses the need for customer asset protection and data availability for distributed environments.

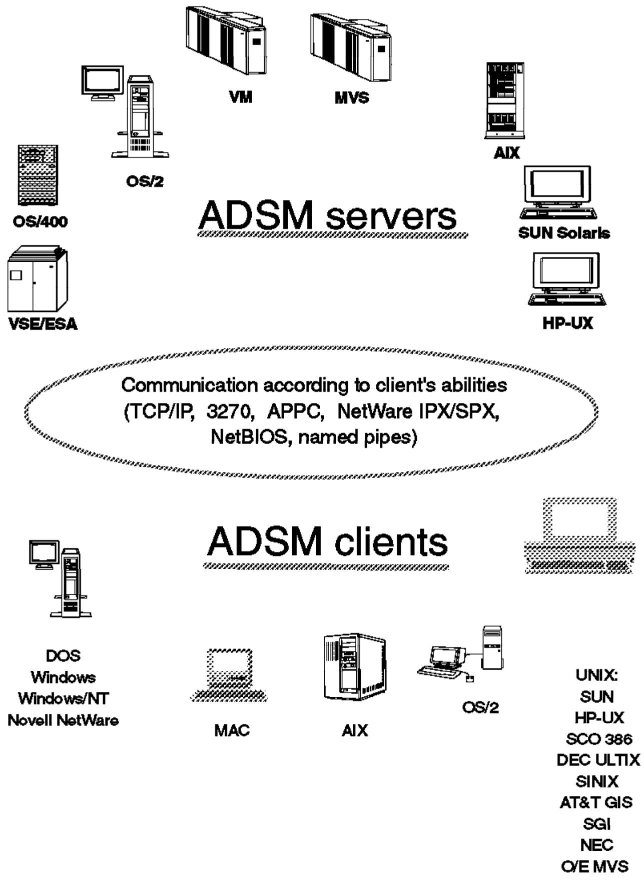


Figure 6. ADSM Platform Support

1.2.3 Main Components of ADSM

Let us look at the main components of ADSM—the backup/archive client, the administrative client, and the server, as shown in Figure 7 on page 13—and then briefly review the application client.

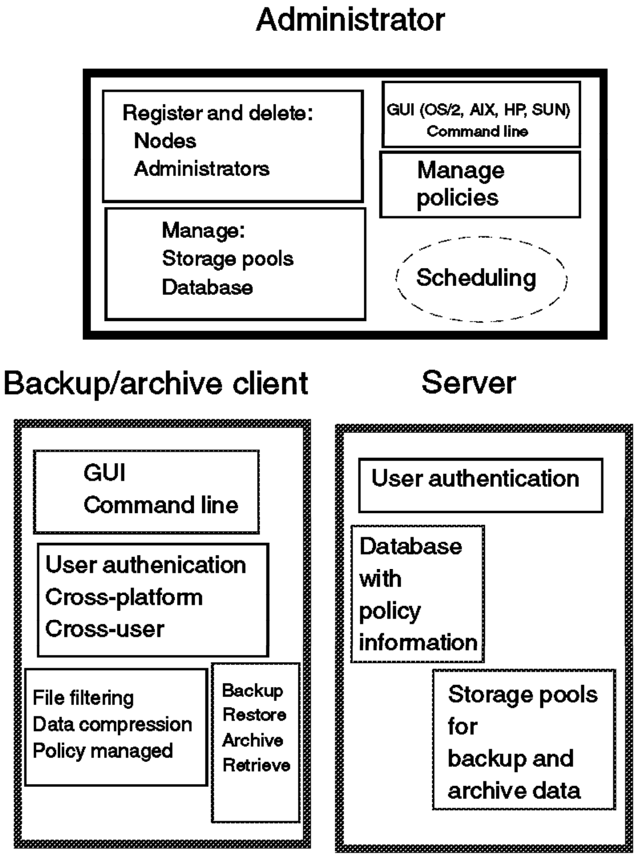


Figure 7. ADSM Storage Management Components

1.2.3.1 Backup/Archive Client

The backup/archive client runs on the workstation and, depending on the platform, provides both a graphical user interface (GUI) and command line interface (CLI). Although all clients are similar, they each have the look and feel of the platform on which they are running. Thus users can back up or restore files using an interface with which they are familiar.

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

Note: The ADSM license distributed with VisualInfo is a limited function license, which does not allow you to perform backup and recovery activities. However, if you purchase a full ADSM license, you can do these activities.

The file compression provided on the client platforms reduces network traffic and the amount of storage required on the server to store the files. However, it can affect ADSM throughput. Typically, a fast processor on a slow line benefits from compressions, but a slow machine on a fast line does not.

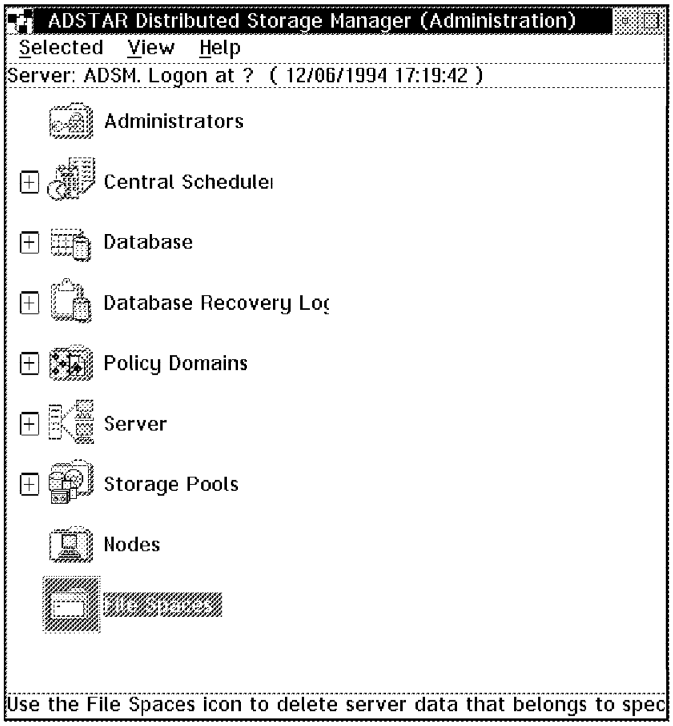
You can specifically include or exclude certain files from being backed up. For example, you might not want everyone to back up their local copies of the OS/2 operating system!

ADSM's *cross-user restore* and *cross-platform restore* provide you with significant flexibility. Cross-user restore enables you to authorize someone else to restore your files. Cross-platform restore enables you to restore your file on a platform different from the platform on which it was backed up. For example, you could back up your file from a DOS workstation, but then restore it to an OS/2 workstation. Cross-platform restores are extremely useful when you migrate to new workstation platforms, or even if you happen to work at a different office one day that has different workstations—you will still have access to the data you backed up.

A separate archive/retrieve function is also part of ADSM. This function provides a way for you to store files that you may not use but that you need to retain for long-term storage. Archive is also useful as a way of reducing the disk space on your workstation. You can archive files for long-term storage and erase the original files from your workstation to create room for more active files and applications.

1.2.3.2 Administrative Client

As shown in Figure 8, an administrator controls or monitors server activity, defines storage management policies for workstation files, and sets up schedules to provide backup and archive services at regular intervals.



ADSTAR Distributed Storage Manager (Administration)

Server: ADSM. Logon at ? (12/06/1994 17:19:42)

Use the File Spaces icon to delete server data that belongs to spec

Figure 8. ADSM Administrative Client

An *administrative client* is a program that allows administrators to control and monitor the server through administrative commands. The administrative program can be installed on a programmable workstation (PWS), personal computer, or mainframe. An administrative client passes commands through an administrative command line. In some cases, a GUI has been added to the administrative client code.

ADSM provides a hierarchical structure to the authority you can grant an administrator. Thus you can establish a flexible an administration scheme while still providing control over your system. The ADSM administrator with overall authority is called the *system administrator.* The other administrators are called *policy, storage, operator, or analyst administrators*, depending on which part of the system they control. Their administrative tasks are separated into logical categories, such as controlling the management policies, the storage pools and databases, the operation of the server, and the analysis of certain server events.

Dividing the administrative authority based on logical categories of tasks is not the only way of granting authority. You can also divide the administrative authority by organization. You can give the logical categories of authority to a department, but only for the data that belongs to that department. For example, you can give a department policy and storage authority for the policy domain and storage pools that it owns.

1.2.3.3 Server

The server component provides storage resources and services for the backup/archive clients. Users can back up or archive their files onto server storage resources, such as disk, tape, or optical devices that are managed and monitored by ADSM server policy.

Figure 9 shows the two key components of the ADSM server: the storage pools where the client files are actually stored, and the database that serves as an inventory or index to the client files within the storage pools. The database consists of the database space and the recovery log. The recovery log keeps track of all changes made to the database.

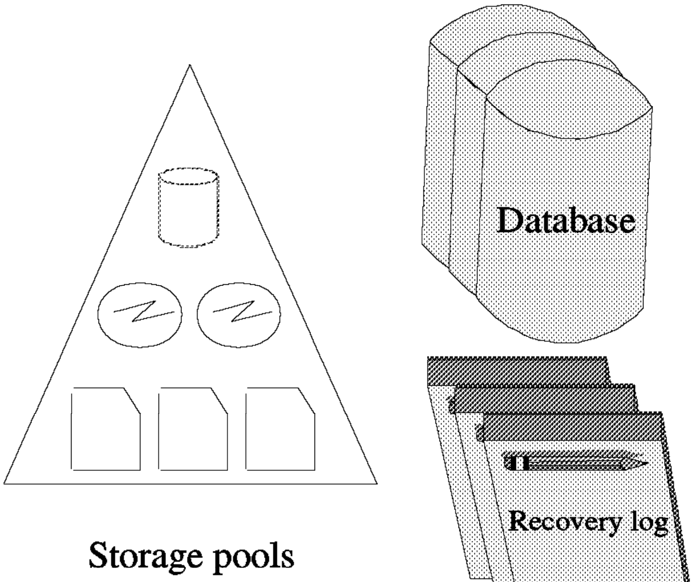


Figure 9. ADSM Server Components

The storage pools contain the client files that have been backed up or archived. A hierarchy of storage media can be used to define the storage pools. The pools can contain disk storage, optical devices, and tape devices. Each ADSM server platform supports a different set of storage media, so you should verify the devices that are supported in your environment.

Data is moved automatically through the storage hierarchy onto less expensive media with ADSM′s migration function. Additional management functions are provided, such as reclamation and collocation for tape management.

The ADSM server is multitasking, so multiple clients can back up data concurrently.

The ADSM database is the heart of the server. The server database is critical to the operation of ADSM because it contains file location information as well as policy and scheduling information. The following information is stored in the database:

- Information about registered client nodes
- Policies assigned to those client nodes
- Schedules and their association with client nodes
- Event records, such as whether a schedule successfully completed
- The activity log that contains the messages generated by the server
- Information about ADSM volumes
- The data storage inventory, that is, the information used to locate files that reside in storage pools

The database has all of the features associated with a database management system. Because the database is critical, many features are built in to ADSM to help maintain the availability, integrity, and performance of the database. Two of these features are the recovery log and mirroring.

A *recovery log* is used to help maintain the integrity of the database. It keeps track of all changes made to the database, so that if a system outage were to occur, a record of the changes would be available in the log. When a change to the database occurs, the recovery log is updated with some transaction information before the database is updated. Thus uncommitted transactions are rolled back during recovery so that the database remains consistent.

Mirroring is the process of writing the same data to multiple storage devices at the same time. The administrator can configure the server so that up to three copies of the database and recovery logs are maintained at all times. This mirroring capability provides nondisruptive and immediate recovery from physical failures on database and recovery log volumes.

If a mirrored volume encounters a media failure, the server automatically places the failing volume offline and continues database operations using the other mirrored copies. Once the failed disk is replaced and made available to the server, it is automatically synchronized with the intact copies.

The mirroring facility improves database performance. The mirrored copies are treated equally; there is no concept of primary copy and alternate copies. Therefore, the server reads from the database copy that is on the device with the best response time.

Another server function, *export/import*, creates a self-describing copy of specified server information. Information that can be exported includes:

- Administrator information
- Client node definitions

- Policy information
- Backup and archive data

Export/import is useful for migration, conversion and workload balancing, as well as cloning of information.

ADSM provides extensive ADSM server database and storage pool backup facilities. Incremental backups are provided as well as a mechanism for offsite backups to aid in disaster recovery.

1.2.3.4 Application Client

The application client is a software application that runs on a workstation and uses the ADSM application programming interface (API) to back up, archive, restore, or retrieve objects from an ADSM server.

As shown in Figure 10, the application client program enables other IBM and non-IBM products to use the storage management services of ADSM. The application client allows applications to back up or archive valuable data in any format that an application programmer specifies.

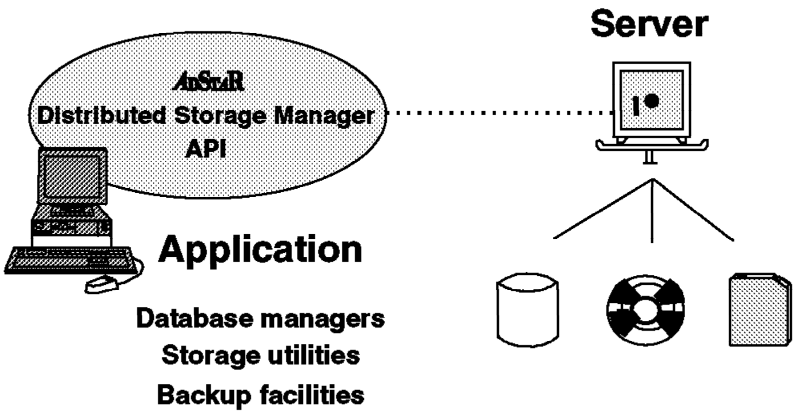


Figure 10. Application Client and ADSM API

The number of ways of using the API is unlimited. You can use it to provide better handling of non-file data in the enterprise, such as databases or image volumes. You could, for example, provide extensions to the existing ADSM back up and restore functions to meet your user′s needs or write a virtual tape device driver so that other applications can use ADSM transparently.

The API is available for the C programming language.

Using this API DB2 family of products enables you to back up databases directly on an ADSM storage pool.

1.2.4 Functions

Let us look at the ADSM backup and restore, archive and retrieve, central scheduling, and policy management functions.

1.2.4.1 Backup and Restore

The *backup* process creates a copy of a client file on the ADSM server, such as the myfile.data file shown in Figure 11. The backup process also backs up the directory in which the file resides.

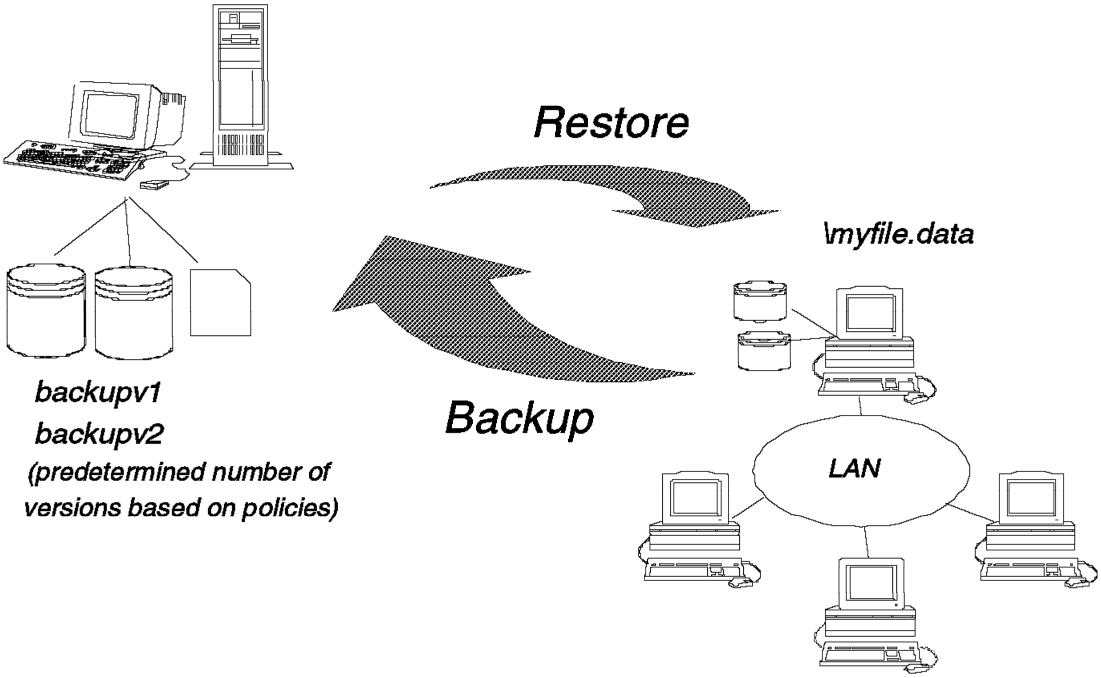


Figure 11. Backup and Restore

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

Selective backup specifies which files a user wants to back up. A selective backup can consist of a single file, or a user can select a directory or subdirectory tree to back up. Because wildcards (*) are allowed in the specification, there is great flexibility in file selection.

The files are backed up according to policies that the administrator has predefined. The policies define, for example, how many backup versions should be retained in the ADSM storage pools, how long to retain those versions, and whether to back up files that are in use. Figure 11 shows that two versions of the myfile.data file are saved in the storage pools.

Restore is the process of copying a backup version from the server to the client. This process is system-assisted; that is, the system performs the restore for the user. The user does not have to call the ADSM administrator to request restoration of the file.

1.2.4.2 Archive and Retrieve

The *archive* process creates a copy of a client file on the ADSM server, such as the myfile.data file shown in Figure 12.

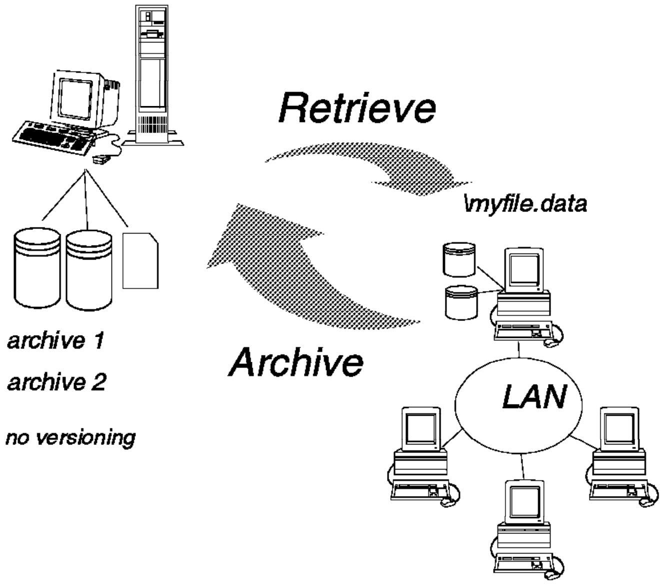


Figure 12. Archive and Retrieve

As with backup, archived files are managed based on policies; however, the archive function does not have a concept of versioning. You can archive multiple versions of a file by invoking the archive function multiple times. In other words, each archived copy is treated as a separate file, not as multiple versions of a single file.

A user can save a description of an archived file so that it is easier to retrieve the file, even if multiple files are archived with the same file name.

Note: The key difference between backing up a file and archiving a file is that the user can erase the original file after archiving it. The archived version is expected to be retained for a long time. Erasing the original file does not affect the retention period for the archived file.

1.2.4.3 Central Scheduling

As shown in Figure 13 on page 20, the ADSM central scheduling facility automates the initiation of client backup, archive, restore, and retrieve, as well as ADSM server administrative operations. It also can schedule any client OS command and ADSM client macros. New clients can be easily associated with schedules in a nondisruptive manner.

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

The administrator is responsible for defining and maintaining the schedules and has the authority to prioritize clients so that clients that contain more important data are given preferential treatment.

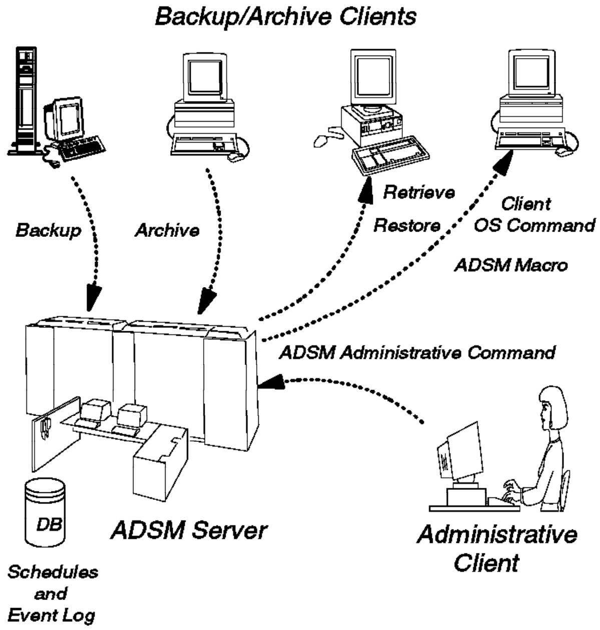


Figure 13. ADSM Central Scheduling

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

Two types of scheduling modes are supported:

- Client polling

  Client polling is supported for all client workstations and all communication methods. With client polling, a client periodically queries (or polls) the server for a scheduled operation and the date and time the operation is to start. The server sends this information to the client. Then the client waits until it is time to start the scheduled operation and executes the operations.

  Operations that can be scheduled are backup and archive. Restore and retrieve cannot be scheduled. Before executing the operation, the client notifies the server that a scheduled operation is starting. On completing the operation, the client notifies the server that the operation has completed, either successfully or unsuccessfully.

  The client initiates client polling by starting the client scheduling program. To start the program the client enters DSMC SCHEDULE. The program will continue to query the server and execute schedules until the user explicitly stops the program or the machine is shut down.

- Server-prompted

  Server-prompted scheduling is supported for all client workstations that use TCP/IP to communicate with the server.

  With server-prompted scheduling, the ADSM client registers its TCP/IP address with the server and then waits to be prompted by the server to begin the scheduled operation. The client then starts and executes the operation.

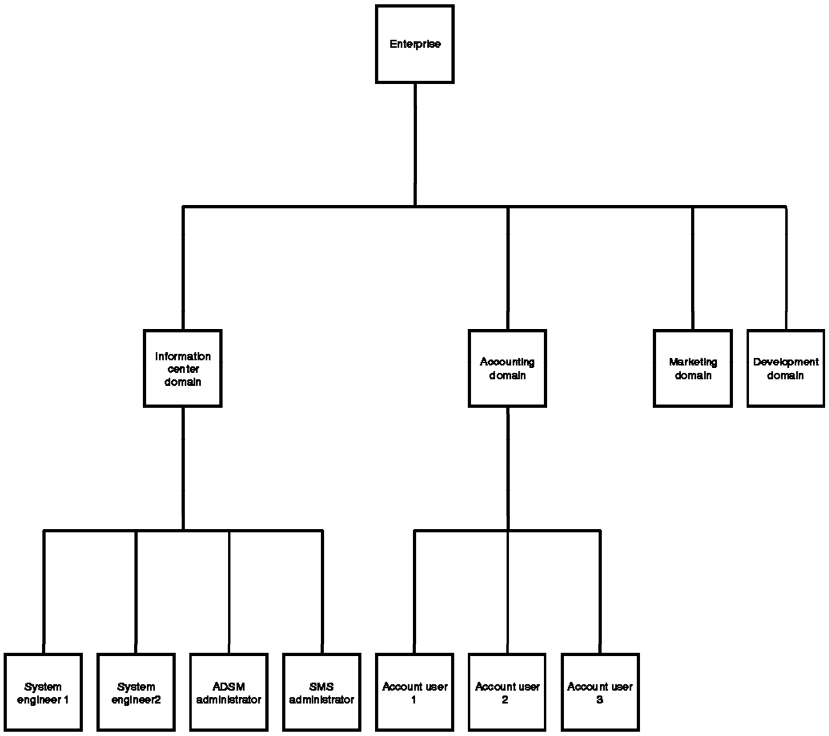
The operation can be backup or archive. Restores and retrieves cannot be scheduled. On completing the operation, the client notifies the server that the operation completed, either successfully or unsuccessfully.

Server-prompted scheduling allows the server to control when clients are contacted to perform a scheduled operation. Server-prompted scheduling maximizes the use of scheduled sessions.

To initiate a schedule, the client must start the client scheduling program by issuing the DSMC SCHEDULE command.

1.2.4.4 Policy Management

As shown in Figure 14, ADSM allows you to manage the backup and archive process based on policies you establish for your enterprise. The granularity of control that you have is down to the file level. You can decide how granular you want your policies to be. You can establish an overall system policy, policies by department or organizational structure, or policies by user or file name. Policy management makes ADSM a true system-managed storage implementation. The elements of policy management are discussed in the following section.



ADSM can be policy managed according to the organization of the enterprise

Figure 14. Policy Management

Policy Domain: A *policy domain* is a group of clients who work according to the same set of policy needs. A policy domain provides a logical way of managing backup and archive policies for a group of client nodes. There is no limit to the number of policy domains that can be defined on an ADSM server.

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

Figure 15 on page 23 shows that clients ACCTREP1, ACCTREP2, and ACCTREP3 belong to the SALES policy domain. Note that schedules also belong to a particular policy domain.

Policy Set: Each policy domain can contain one or more policy sets. A *policy set* contains one or more management classes. A policy domain can have more than one policy set, but only one policy set can be activated at any point in time. Each policy set contains a default management class and can contain any number of additional management classes. Policy domain and policy set information is stored in the server database.

Figure 15 on page 23 shows three policy sets, with an active policy set that contains two management classes.

Management Class: Policy sets contain one or more *management classes*. Management classes contain a backup copy group, an archive copy group (not used by VI or DL), or no copy group. You can think of management classes as a “service level agreement” that you have with your clients on how their backup and archive data is to be handled. There is a concept of binding the management class to the file when it is backed up or archived. Thus the management class is associated with that file. You can rebind a file with a new management class. Users can either use the default management class or explicitly select a management class that is within the active policy set to which they have access. Figure 15 on page 23 shows two management classes, MC1 and MC2. Management class MC1 contains both a backup and archive copy group; MC2 contains only an archive copy group.

Copy Group: Copy groups are where you specify the parameters that will control the generation and expiration of backup and archive data. There is a separate copy group for backup and one for archive. In the current ADSM product, all copy groups are named STANDARD. Again, copy group information is stored in the ADSM server database. Let us look at the parameters that you can use to control your backup and archive data. Remember that the span of control is at a file level.

Note: An archive group is not used by VisualInfo or Digital Library.

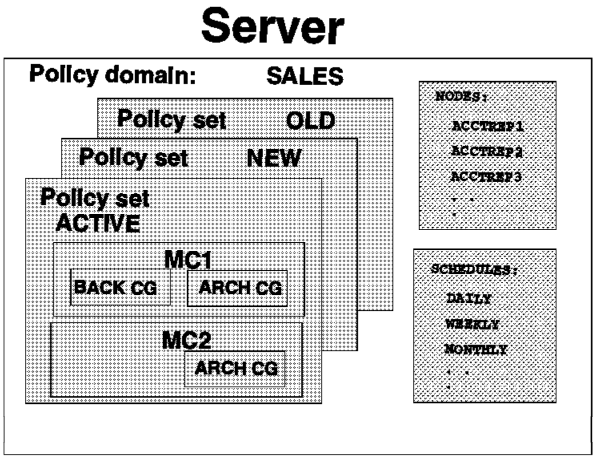


Figure 15. Policy Management Elements

Both backup and archive copy groups have similar parameters except that there is no concept of versioning with archived files. Let us look at each parameter shown in Figure 16.

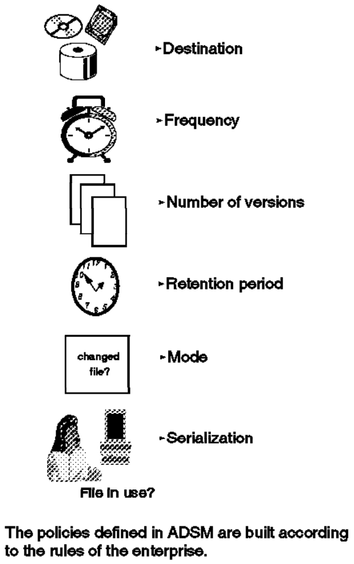


Figure 16. Copy Group Parameters

Destination specifies the name of the storage pool where the server stores the backed up or archived files.

Frequency for a backup file specifies the minimum number of days that must elapse between incremental backups. This parameter is not used for selective backups. Frequency for an archive file is always command (CMD). A file is only archived when a client issues an archive command or chooses archive from the GUI.

The concept of *versioning* applies only to backup files. You can specify two different parameters to tell ADSM how many versions of a backup file you want it to maintain.

- The Version data exists parameter specifies the maximum number of different backup versions the server retains for files and directories that currently exist on the client workstation. The most current backup version is called the active version. All other versions are called the inactive versions. When the maximum number of versions is exceeded, the server rolls off the oldest version.
- The Version data deleted parameter specifies the maximum number of different backup versions the server retains for files and directories that have been erased from the client workstation.

The *retention period* parameter specifies how long to retain the backed-up and archived files. There are two retention parameters for backed up files that correspond to the two types of versioning, and there is one retention parameter for the archived files.

- Retain extra versions specifies how many days the server retains the inactive backup versions.
- Retain only version specifies how many days the server retains the backup versions it has of a file when the original file has been deleted from the workstation.
- Retain version specifies the number of days an archived copy remains in data storage.

With the *mode* parameter, you can specify file backup depending on whether the file has changed since the last backup. This parameter applies to incremental backups, not selective backups. The options for mode are modified and absolute.

- Modified means that you want to back up the file only if it has changed.
- Absolute means that you want to back up the file regardless of whether it has changed.

  For archive files, the mode is always absolute.

Serialization specifies how files or directories are handled if they are modified during the backup or archive process. The serialization parameter has four options: static, shared static, shared dynamic, and dynamic.

- Static specifies that if a file or directory is modified during the backup or archive process, ADSM will not back up or archive the file. The static mode is not supported on the DOS platform.
- Shared static specifies that ADSM will retry the backup operation as many times as specified in the client′s option file. The default is four retries. If the file or directory is modified during each backup or archive attempt, ADSM will not back up or archive the file.
- Shared dynamic specifies that if a file is modified during a backup or archive attempt, ADSM will only back up or archive the file on its last retry.
- Dynamic specifies that even if the file is modified during the backup or archive attempt, ADSM will back up or archive the file anyway. No retries are required.

1.2.5 Advantages

IBM′s commitment to provide storage management functions for remote systems and multiple platforms will enable the enterprise to gain substantial benefits. Let us conclude this section of the book by summarizing the many advantages of ADSM. We have grouped them into three categories: cost reductions, increased productivity, and increased security of corporate assets.

1.2.5.1 Cost Reductions

As proven in the large systems arena, fewer people will need to be involved in the process of storage management. Because ADSM provides a single approach to storage management for your entire enterprise, perhaps all enterprise storage can be managed by one central group.

Available storage is managed more effectively because inactive data is moved to other media, thus increasing the amount of fast access disk space required for active data.

Distributed data growth is easily monitored and controlled because less duplication of data is likely to occur as users become confident that their data is secured. ADSM can also help you avoid backing up multiple copies of system software.

1.2.5.2 Increased Productivity

With storage management across an enterprise, any issues regarding data integrity can be restored by the system more quickly and accurately, while adhering to enterprise-defined standards.

Many tasks that people do manually can now be automatically managed by fewer people. End users or system administrators from different systems and platforms will spend less time on availability and space management. They will experience fewer losses of critical data and fewer out-of-space incidents. They are freed from the tedious manual work of managing the disks they own; data management can be done transparently, with little impact on their usual work.

Using a single storage management strategy across your heterogeneous platforms encourages network-wide standards. This might be at a high level, where standard policies of how to manage the data are established, or at a lower level, where, for example, file naming conventions become more standardized to improve restorability to other platforms.

1.2.5.3 Increased Security of Corporate Assets

With a structured process for backing up your data in your enterprise, you can be assured that your data is available when you need it. The risk of losing valuable data, which is a valuable corporate asset, has been substantially reduced.

You may also be taking advantage of more reliable server storage than that provided on your local workstation. The system on which the server storage resides may also have a higher level of security enforcement and control. The server storage might provide a higher level of security simply by being in a central data center (glass house) that requires a badge lock to enter.

Now that we have given an introduction to VisualInfo and ADSM, let us look at the system managed storage component of VisualInfo (and Digital Library) in more detail.

Chapter 2. SMS Concepts and Components

System Managed Storage is a very powerful concept, but you must understand its terms to exploit it to the fullest. In this chapter, we explain the concepts involved in the configuration of SMS.

2.1 Why Do You Need to Manage the Storage?

The main purpose of SMS is to optimize the use of the different kinds of storage you can have in your system, and to use that storage to match your document needs. Those needs are defined by how you use your documents in your business. You can have many different types (index classes) of documents, but many of them have the same storage needs. What defines the storage needs of a document are the *time* the end user must wait to get it and the *cost* associated with the storage media.

If you ask an end user “What is an acceptable wait time for a document?” the answer is “Less than a second.” That is because users are accustomed to the response times of transactional systems. But if you ask them “How much time does it take to get a document from your paper archive?” the answer is “From 30 minutes to 24 hours.” With content management systems (such as VisualInfo and Digital Library) we try to close this huge gap in access times by using SMS. We achieve this goal with intelligent use of the storage systems.

The question SMS tries to answer is “Where should a document be stored for each phase of its life to optimize the access time and the cost of the media?”

Our objective is to reduce the wait time for the most-used documents to something approaching one second and, for the archived ones, to less than 60 seconds, while staying within budget!

2.2 Types of Storage

Content management systems have to store very large objects. A scanned 20 page document can be 1 MB, and a 20-second video clip can be dozens of megabytes. Those systems must support inline, online, and offline object archiving to multiple storage media.

Storage media are grouped into three main classes: solid state, magnetic, and optical. Solid state, or Random Access Memory (RAM), is the most expensive and the fastest. RAM access time is measured in nanoseconds; however, it has limited capacity.

Magnetic media are tape or disk. Magnetic disks are by far the most common storage devices in use today. The disks vary in capacity from a few hundred megabytes to gigabytes. Magnetic disks are random access media, in that any portion of the disk can be accessed at any time; but magnetic tape is a linear media. Data contained near the end of a tape can only be accessed by fast forwarding the tape to that section and then reading it. Depending on the format, magnetic tapes can deliver capacities between 1GB and 50GB per tape. Magnetic tapes are slower than magnetic disks, but are cheaper.

The third kind of storage is optical disk technology, which is in widespread commercial use. Optical disks are available in 3.5, 5.25, 12, and 14 inch formats. Because no mechanical portion of the disk drive mechanism ever makes physical contact with the optical disks, they offer greater data reliability. They are less susceptible to the data loss that happens in magnetic disks when the drive mechanism accidentally crashes into the disk surface. According to some estimates, optical disks may reach capacities of 20GB per surface by the turn of the century. Optical disks should be considered when you need a very long term storage, such as for historic documents. In some countries, documents stored on Write Once Read Many (WORM) optical disks have legal value, so that is the only media you can use if you need to store documents for legal reasons.

Deciding what kind of storage to use is not easy. Prices change daily. DASD, which used to be very expensive, is now almost in the same cost per GB ratio as optical disks. You have to study carefully all of the available options, but remember these three points:

- Magnetic devices are more susceptible to damage than optical devices. If you need long term storage, it is better to use optical.
- Content management systems are used to store objects for a long time, from one year to 100 years and more. (Digital copies of objects in museums should last forever.) When you select your storage provider, ask for long term support. It is quite uncomfortable to have 500GB of documents stored (representing many months of hard work), then receive a note about the disappearance of the company that maintains your system and supplies you with tapes or disks.
- Select standard devices and formats. They guarantee portability between devices and providers.

2.3 Device Managers

If you use any kind of storage in your VisualInfo system, you will need two software components:

1. The operating system device driver
2. The VisualInfo device manager

The operating system device driver is provided either by the operating system itself or by the storage device manufacturer. Without it, the operating system cannot “see” the device.

The VisualInfo system device manager is provided by VisualInfo. A *device manager* is the interface between the device drivers and VisualInfo applications. These are the device managers for AIX and NT that come with VisualInfo:

- FRNNHDDM: Windows NT fixed disk device manager
- FRNXHDDM: AIX fixed disk device manager
- FRNNADDM: NT ADSM device manager
- FRNXADDM: AIX ADSM device manager
- FRNXMADM: AIX media device manager (VideoCharger)

The ADSM device managers are used for all the devices supported by ADSM: optical disks, optical libraries, and tapes.

2.4 Storage Classes

Now we have the devices recognized by the operating system, and we have the VisualInfo software (device managers) that handles those devices. We say that all the devices that have similar properties belong to the same Storage Class.

Those properties are:

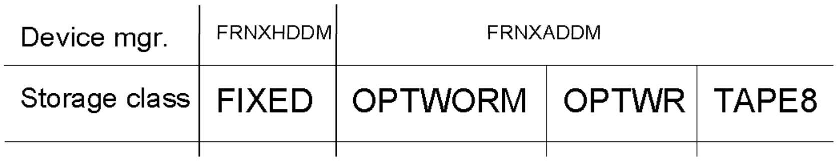
- Access time
- Capacity
- Rewritability
- Data format
- Whether volumes are removable

For example, we can have one storage class for optical WORM disks and another for rewritable optical disks. Both classes are managed by the same VisualInfo device manager, but they have different properties.

As previously mentioned, documents have different storage needs during their lives. We can move documents from one device to another to satisfy those needs by moving (migrating) them from one storage class to other. When defining storage classes, remember these two points:

- Each storage class has different properties: access time, capacity, and so forth.
- Documents are migrated from one storage class to another

Figure 17 represents those concepts in a chart:



Device mgr.	FRNXHDDM	FRNXADDM		
Storage class	FIXED	OPTWORM	OPTWR	TAPE8

Figure 17. SMS Storage Classes

In this AIX example, we have four storage classes:

FIXED Very good access time, non-removable volumes, limited capacity (<20GB). Implemented with DASD, and using the fixed disk device manager FRNXHDDM.

OPTWORM Medium access time, removable volumes, not rewritable, high capacity (600GB). Implemented with optical libraries.

OPTRW Medium access time, removable volumes, rewritable, high capacity (600GB). Implemented with optical libraries.

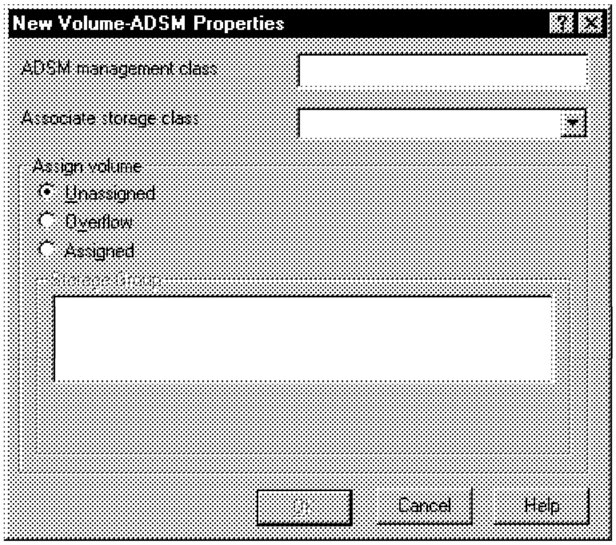
TAPE8 Slow access time, removable volumes, unlimited capacity. Implemented with an 8mm tape drive.

The last three classes use the ADSM device manager FRNXADDM.

2.5 Volumes

Now all the storage devices have been classified into storage classes, ′ but where are the real files stored? We do not store objects into the storage class FIXED, but onto the C: drive or in the /usr/lpp/ directory. In SMS terminology, we say that we store objects into volumes. A *volume* is a physical storage medium that a storage class manages using activities such as read and write. On Windows NT, a volume can be a DASD partition or an ADSM management class; on AIX it is a DASD filesystem or an ADSM management class.

Note that there is a naming conflict: an ADSM management class *is not* the same as an SMS management class (which is discussed later). When you define a new ADSM volume to VisualInfo, you must specify the name of an *existing* ADSM management class (see Figure 18).



ADSM management class:

Associate storage class:

Assign volume
● Unassigned
○ Overflow
○ Assigned

Storage Group

Figure 18. New ADSM Volume Window

When you click on **OK**, the SMS server starts a conversation to the ADSM server and queries it about the existence of the management class. If the name of the ADSM management class is, for example, STANDARD, the name of the VisualInfo ADSM volume will be ADSM:STANDARD.

Figure 19 shows an example:

Device mgr.	FRNXHDDM	FRNXADDM		
Storage class	FIXED	OPTWORM	OPTWR	TAPE8
Volumes	F1 F2 F3	OR01 OR02	OPTRW1	TAPE1

Figure 19. SMS Volumes

Here is the description of each volume:

- F1, F2 and F3 are DASD volumes (storage class FIXED). On NT they can be three different partitions or three different disks; on AIX they can be three different filesystems.
- OR01, OR02 are two VI_ADSM_volumes of class OPTWORM. Following are some reasons for having two volumes:
  1. You want to associate some disks in the optical library to one volume and other disks in the same optical library to the other volume. For example, you need to store some documents for 5 years and others for 20 years, and you want to keep them in different groups of optical disks.
  2. You have two different optical libraries, and you associate each volume with one of them. For example, you start with one optical library and you add a new one to the system because you need more space.
- OPTRW1 is a VI_ADSM_volume of class OPTWR. This volume is an ADSM-managed optical library with R/W drives.
- TAPE1 is a VI_ADSM_volume of class TAPE8. It is an ADSM-managed 8mm tape drive. You can have many physical tapes associated with this drive. Each one is an ADSM volume with its own name (this is handled by ADSM). For VisualInfo, the volume TAPE1 is considered to be one unique volume. VisualInfo does not care about mounting tapes; that is the job of the ADSM server.

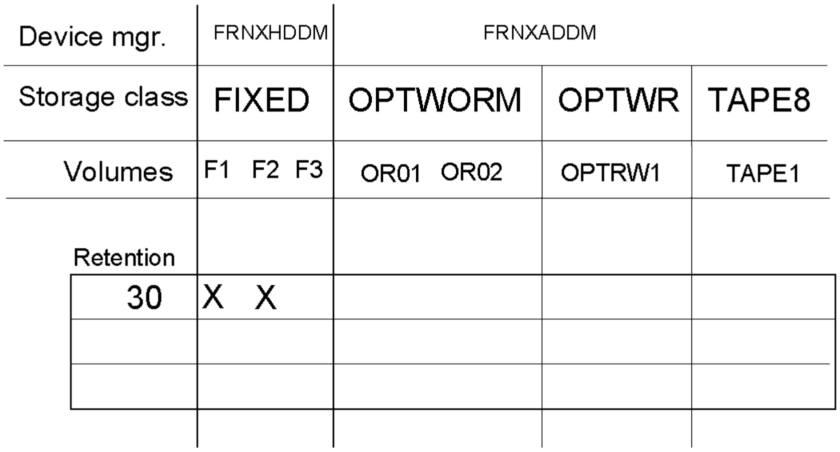
2.6 Management Classes

Migration is the key concept of SMS. When objects are moved from one device to other, you say that they are *migrated* from one storage class to another. All you have to do is define how many days the objects will stay in each storage class. That time is called a *retention period*. A management class is simply an ordered list of values for retention period and storage class.

But you can have more than one volume in one storage class. In fact, you have three volumes in the class FIXED in our example, and two volumes in the class OPTWORM. Sometimes you need to define the volumes for a specific kind of document. For example, the volume OR02 can be a newer and faster optical library than OR01, and you want to select one of these volumes when you decide to migrate an object to the OPTWORM storage class.

Our charts can help us in defining management classes and volumes. Suppose you have some documents that must stay 30 days on DASD, 100 days on optical R/W disks, and one year on tape:

- Write the retention period on the left column and select the volumes you want to use for the first class:

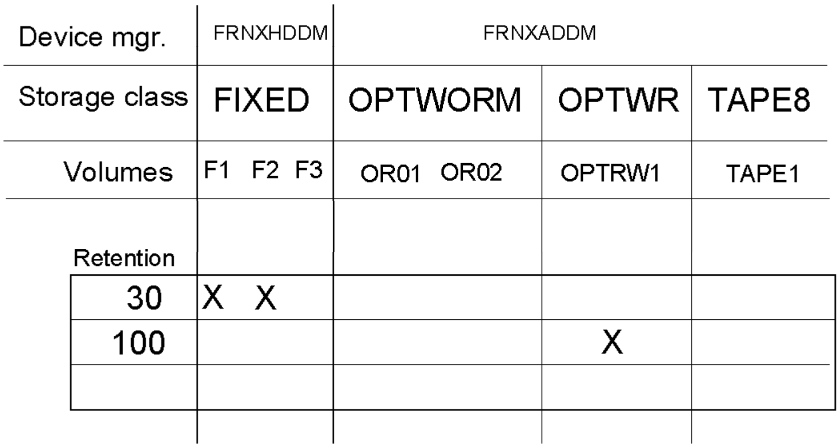


Device mgr.	FRNXHDDM	FRNXADDM		
Storage class	FIXED	OPTWORM	OPTWR	TAPE8
Volumes	F1 F2 F3	OR01 OR02	OPTRW1	TAPE1
Retention				
30	X X			

Figure 20. Management Class MCLASS_A, 1 of 3

We have selected volumes F1 and F2 from class FIXED, with a retention period of 30 days.

- Now add a second row, write the retention for the next class and select the volumes:

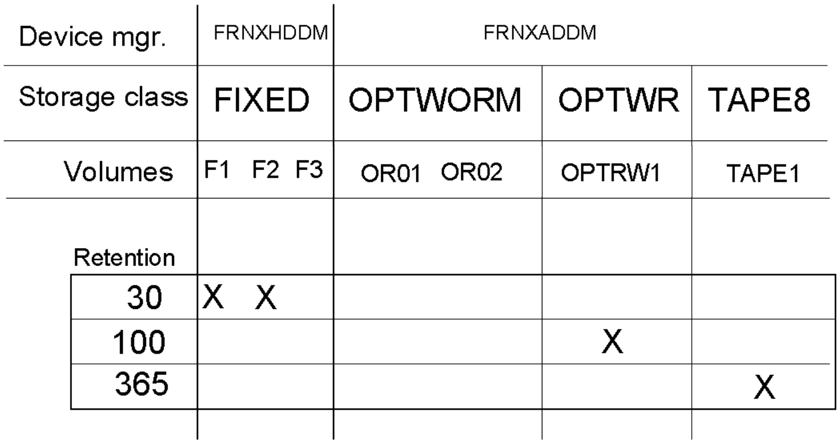


Device mgr.	FRNXHDDM	FRNXADDM		
Storage class	FIXED	OPTWORM	OPTWR	TAPE8
Volumes	F1 F2 F3	OR01 OR02	OPTRW1	TAPE1
Retention				
30	X X			
100			X	

Figure 21. Management Class MCLASS_A, 2 of 3

We have only one volume in class OPTRW, so we select it and specify a retention of 100 days.

- Finally, add a third row for the last class:



Device mgr.	FRNXHDDM	FRNXADDM		
Storage class	FIXED	OPTWORM	OPTWR	TAPE8
Volumes	F1 F2 F3	OR01 OR02	OPTRW1	TAPE1
Retention				
30	X X			
100			X	
365				X

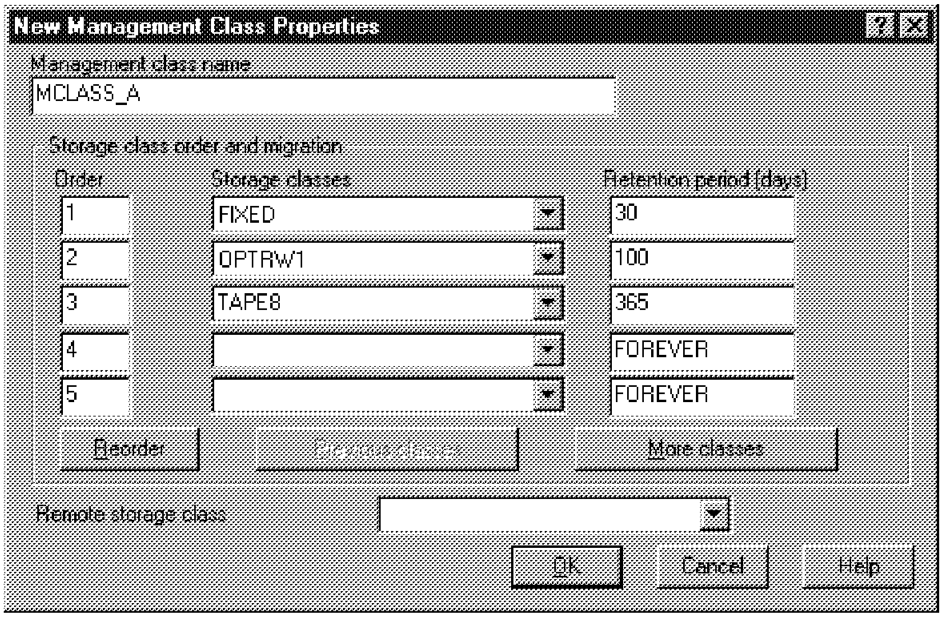
Figure 22. Management Class MCLASS_A, 3 of 3

We select volume TAPE1 of class TAPE8, with a retention period of one year.

Note: Remember that in the same row, you can select volumes from only *one* management class.

Note: These volumes will be selected when you create a Storage Group.

To implement this example in a real system, we use the System Administration program, and we write these values in the New Management Class Properties window:



New Management Class Properties

Management class name: MCLASS_A

Storage class order and migration

Order	Storage classes	Retention period (days)
1	FIXED	30
2	OPTRW1	100
3	TAPE8	365
4		FOREVER
5		FOREVER

Figure 23. New Management Class Window

When the last retention period is reached, VisualInfo does not automatically delete the object. You must do this yourself, either manually or by writing a program which will delete objects which match your deletion criteria. See page 41 for more information.

This way of defining management classes comes from the days when the storage devices were managed directly by VisualInfo. In fact, this is the way we do it on the OS/2 platform. But, on NT and AIX, the storage devices are managed by ADSM except for magnetic disk, which has the FRNxHDDM device manager and is managed directly by VisualInfo.

The use of ADSM has a deep impact on the definition of management classes and on the concept of migration. We can use a system with the management class we have just defined in the preceding example, but it would be an inefficient system. This is because, when the object server migrates an object from optical to tape, it has to recover the object from optical and send it to tape, which is a time-consuming task. It is better to give to ADSM the job of migrating objects between storage classes. In our example, the process would be:

1. VisualInfo stores the object on fixed disk.
2. VisualInfo sends the object to the ADSM:OPTRW1 volume after 30 days.
3. ADSM moves the object to tape according to the policy defined in the storage pool hierarchy.

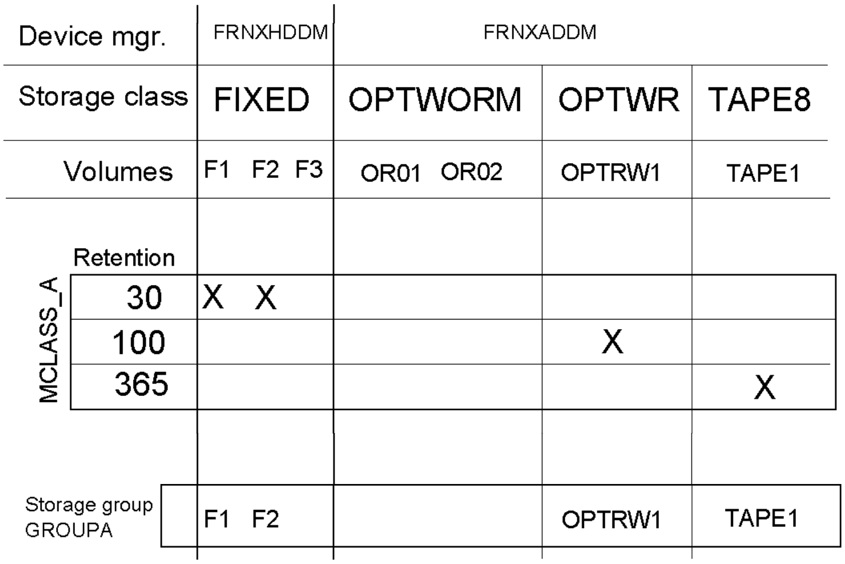
In this process, there are two key concepts you must remember:

- When an object is moved to ADSM, VisualInfo does not know if the object is on optical or tape. All VisualInfo knows is that the object is someplace inside the ADSM system.
- The migration inside ADSM is based on *occupancy*, not time. That means that an object will be moved to tape when the optical volume is n% full, not after n days on that storage class.

For VisualInfo, there are only two levels in the storage hierarchy: fixed disk and ADSM volumes. It is the ADSM server that is responsible for the migration to optical disks and tapes.

2.7 Storage Groups

A *storage group* is the set of all the volumes that are used by one management class, as shown in Figure 24 on page 35.

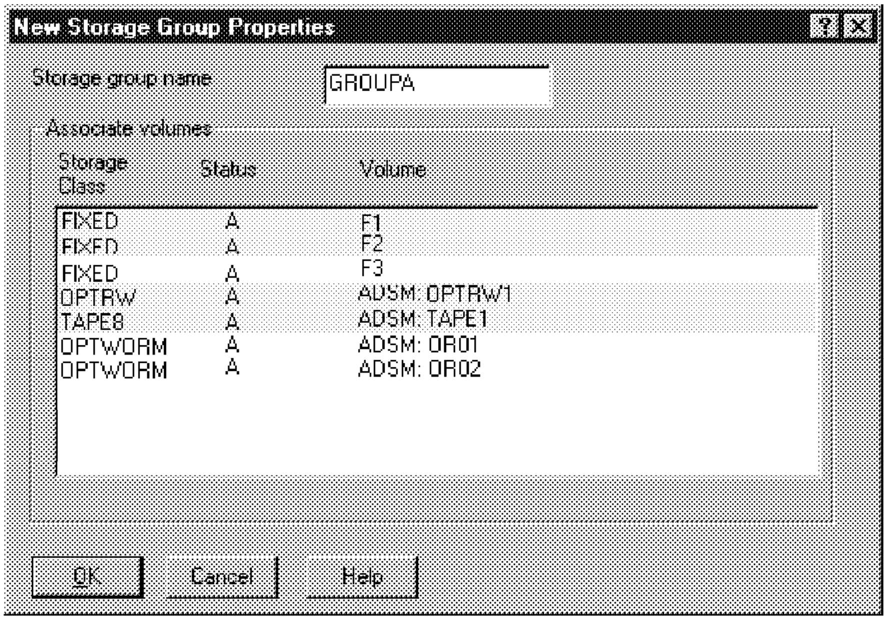


Device mgr.		FRNXHDDM	FRNXADDM		
Storage class		FIXED	OPTWORM	OPTWR	TAPE8
Volumes		F1 F2 F3	OR01 OR02	OPTRW1	TAPE1
	Retention				
MCLASS_A	30	X X			
	100			X	
	365				X
Storage group GROUPA		F1 F2		OPTRW1	TAPE1

Figure 24. Storage Group

At the bottom of the chart, we write the names of the volume that we chose for the management class, and we give a name for the group.

To define a storage group with the System Administration program, we just write the name of the new storage group and select the volumes we want to include in the storage group. See Figure 25.



New Storage Group Properties

Storage group name: GROUPA

Associate volumes

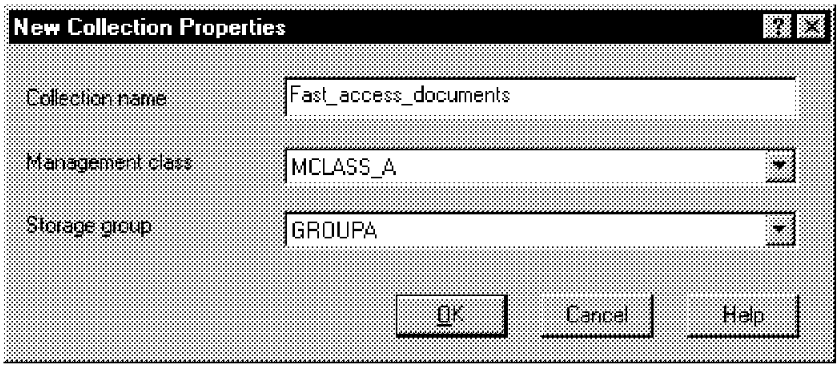
Storage Class	Status	Volume
FIXED	A	F1
FIXED	A	F2
FIXED	A	F3
OPTRW	A	ADSM: OPTRW1
TAPE8	A	ADSM: TAPE1
OPTWORM	A	ADSM: OR01
OPTWORM	A	ADSM: OR02

Figure 25. New Storage Group Window

2.8 Collections

A *collection* is an association of a management class and a storage group. It is what we use to link a document type (index class) with SMS. Each document is stored in a collection. This collection controls which volumes may be used for a storage class (by checking the storage group), and for how long (by checking the management class).

The chart in Figure 24 on page 35 represents a collection, and it has all the information related to it. To create a collection with the System Administration program, use the window shown in Figure 26:



Collection name: Fast_access_documents

Management class: MCLASS_A

Storage group: GROUPA

Figure 26. New Collection Window

Write the name and select the management class and the storage group from the lists.

2.9 Default Collection and Object Server

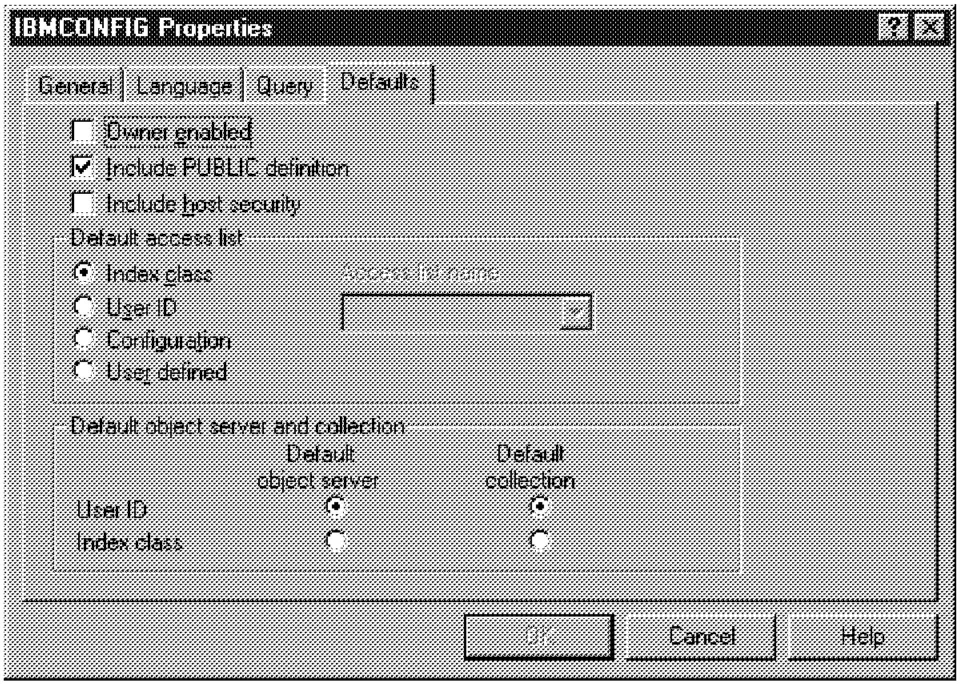
Remember that each document in your system must be associated with a collection, because collections have the information the system needs in order to know when and where to store documents.

When you index a document in your system, you can assign it a collection in two different ways:

1. Use the SimLibCatalog Object API to specify the collection when cataloging the object. This is done by filling the fields of the SMS structure. See Appendix A, “Updating SMS Parameters through APIs” on page 189.

2. Use the system-assigned default values for the default object server and collection. This default value is a setting in the Defaults tab of the library server configuration properties (see Figure 27 on page 37).

   The following screens show the default values using the Windows NT Administration client.



□ Owner enabled

☑ Include PUBLIC definition

□ Include host security

Default access list

● Index class

○ User ID

○ Configuration

○ User defined

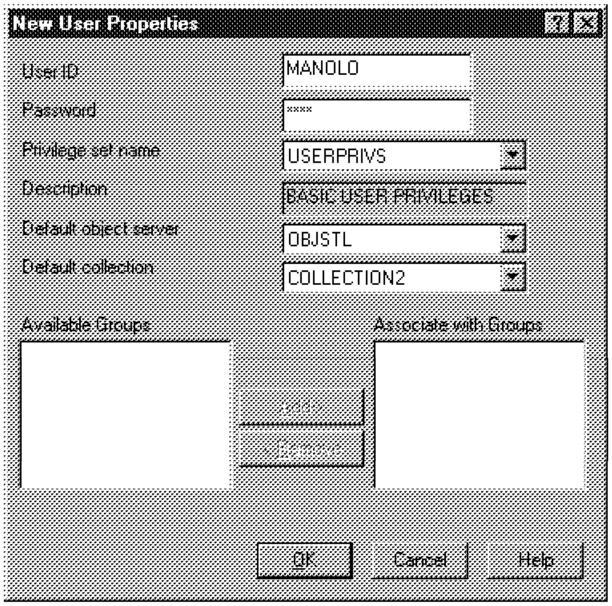
Access list name

Default object server and collection

	Default object server	Default collection
User ID	●	●
Index class	○	○

Figure 27. Default Collection

If you allow the system to assign the default collection, then you have to define the default values in the user properties and in the index class properties (see Figure 28):



User ID: MANOLO

Password: ****

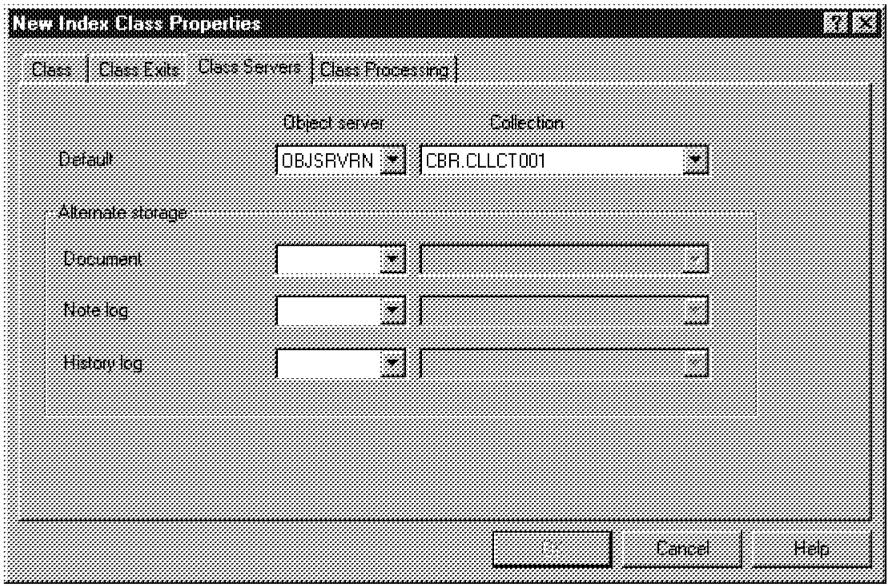
Privilege set name: USERPRIVS

Description: BASIC USER PRIVILEGES

Default object server: OBJSTL

Default collection: COLLECTION2

Figure 28. User Default Storage



Object server | Collection
Default | OBJSRVRN | CBR.CLLCT001

Figure 29. Index Class Default Storage

In your system you can have many object servers, and you can set those values to achieve many different goals. To explain all the possibilities is beyond the scope of this book.

2.10 Migration Processes

Now that you have reviewed the basic SMS concepts, it is time to have a look at the SMS internals (see Figure 30).

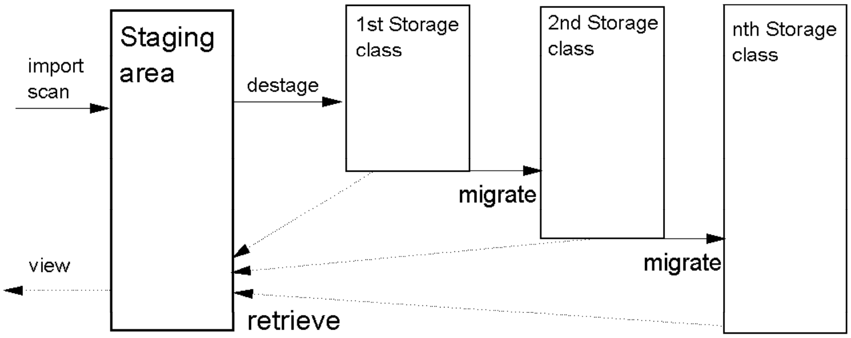


Figure 30. SMS Processes

When an object is initially stored in the system (either with a scanner or with some import program), the object server gets the object directly from the client and writes it into the staging area. The staging area is a critical component of the object server. It is defined at installation time, and it resides on DASD. When first stored in the staging area, the object/file has a file attribute of R for read only to prevent inadvertent deletion or changing of the file. The file name assigned to the object is:

```
collection ID as five digit string + object name
```

The object name for objects created by VisualInfo is dynamically generated by the library server and not stored in a table in the library server. The name format is:

```
first eight characters from itemid in SBTITEMS or SBTPARTS +
″.″ +
last eight characters from itemid +
REPTYPE field from SBTPARTS (usually ″FRN$NULL) +
″.P″ +
part number as hex number, from PARTNUM in SBTPARTS +
″.V″ +
version number, from VERSIONNUM in SBTPARTS
```

SBTPARTS in a library server database table (see *VisualInfo Programming Reference Volume 2* for a description of all the library and object server tables). The current release of VisualInfo does not support versioning, so the last characters would be “.V0” or “.V1.” Here is an example of one of those filenames:

```
00001.RYWOH2EF.QNH5JNX1.FRN$NULL.P1.V0
```

At this time a row is added to the BASE_OBJECTS table (an object server database table) for this newly added object, and the row will contain:

- The object′s name (key field)
- The collection ID
- An actiondate of 12/31/9999 (varies according to the current date and retention period)
- A null filename
- A create date and reference date of the time of storage
- A status of S meaning that the object is currently in the staging area
- A path of 0
- Management class and storage class IDs corresponding to the object′s collection
- Dates for each class indicating when that object first entered that class

The management of the staging area is controlled by the *destager* process, which moves new and modified objects to their first SMS storage class, and the *purger* process, which can delete objects that have been destaged. Parameters that specify the size of the staging area, the time between startup of the destager and purger processes, and the percentage of used space to reclaim, are set through the system administration application.

When the destager runs, the file is renamed and copied to the first storage class defined in the corresponding management class. You have two possibilities on the NT and AIX platforms (OS/2 has others):

1. You can copy the file to a VisualInfo-managed DASD volume. In this case the file is stored in an area named LBOSDATA (ubosdata for AIX) which

resides on a previously defined SMS logical drive for NT or OS/2 (filesystem for AIX). It is copied to a subdirectory within LBOSDATA which corresponds to the object′s collection ID and “bucket” path. The OBJ_PATH field in the BASE_OBJECTS table corresponds to the bucket in which this object resides. The number of buckets a volume can have is in the BASE_VOLUMES field named VOL_NUMBUCKETS. This tells how many subdirectories underneath the collection ID subdirectory to divide the data among. This field cannot be updated through the SMS system administration program and has a default value of 1.

Hint

To update this field, connect to the object server database and execute this SQL command:

```
UPDATE BASE_VOLUMES SET VOL_NUMBUCKETS=x WHERE VOL_VOLUMEID=n
```

where “x” is the new number of buckets, and “n” is the ID of the volume.

Therefore, a file which is stored on a volume whose collection ID is 6 and for which the volume has a numbucket of 2 may be referenced like this: E:LBOSDATA0000601filename.xxx, where the 01 refers to the second bucket (00 is the first) and E is the drive letter for the referenced volume.

2. You can copy the file to an ADSM-managed volume. In this case the object server starts a session with ADSM, sends the object, and closes the session (see Chapter 3, “VisualInfo and the ADSM Interface” on page 49).

Before copying the file to a VisualInfo-managed DASD volume, the destager checks for free space against the file′s size. If there is not enough space, an error message is sent. The file may be written to another volume in the storage group. But when copying to an ADSM-managed volume, it does not check for free space because it considers the ADSM volumes as infinite in size.

At this time the object′s row in the BASE_OBJECTS table is updated to reflect the new volumeid, filename, path and actiondate. Its status will now change from S to A, meaning the object has been archived. The archive filename on the volume is composed of random characters unrelated to the object name. You cannot tell by looking at the file what object it contains, as there is no VisualInfo-supplied header on the object. The only mapping is through the BASE_OBJECTS table. The OBJ_ACTIONDATE is updated to today′s date plus the MGP_SINCEENTER value for this storage class within this management class′ transition (BASE_MGTTRANSITION table).

The file which is in the staging area is now eligible for purging. The destager will set the file attribute accordingly (delete the rw attribute), so that when the purger runs the file can be deleted from the staging area. The purger deletes objects according to their last reference date, so objects most recently used are less likely to be deleted.

The *migrator* process moves the objects from one storage class to another according to the policy specified in the management class. When the migrator runs, it checks the OBJ_ACTIONDATE field and if this date is equal or less that today′s date, it moves the object to the next storage class. At this time the object′s row in the BASE_OBJECTS table is updated to reflect the new volumeid, filename, path and actiondate.

If the new storage class has a retention period of FOREVER, then the OBJ_ACTIONDATE is updated to 12/31/9999. This means that if you update the management class and add a new storage class, this object will never be migrated (or at least not for 8,000 years).

The migrator deletes objects that have been archived. A delete (discard) request from an application does not cause the immediate deletion of an archived object. The object server marks the object as discarded (OBJ_STATUS=′D′) and deletes any staged copy of the file. Nobody can retrieve the object once it has been marked as discarded. On the next migration cycle, the migrator will delete the actual object and the row from BASE_OBJECTS. This delayed action means the delete request does not have to wait to mount the volume with the object.

The purger and destager wake up according to their cycle times. They go to sleep for the cycle time after they finish running. The cycle time is not added to the time they last ran. For example, if the destager ran at 1:00pm for 10 minutes and had a cycle time in BASE_CNTL of 15 minutes, it would run again at 1:25pm, not 1:15pm. VisualInfo does not support exact scheduling of the purger and destager. However, it does support exact scheduling for the migrator.

2.11 Remote Object Servers

One of the most powerful SMS components is the *remote* object server. All discussion in previous sections applies to objects in the same object server. But in the bottom field of Figure 23 on page 33 is a field called Remote Storage Class. If you fill in this field, then when the document finishes its life in the last storage class of that management class (in the example, when it stays one year in TAPE8), it is moved to the remote storage class that has been defined on another object server.

When the document is moved to the remote storage class, it starts a “new life” in that remote object, and is assigned to a new collection.

This process can be used when you migrate from one platform to other, for example from NT to AIX or from OS/2 to MVS. After migrating the library server database, you define remote storage classes and all the objects from one object server are moved “automagically” to the new object server.

If you have a system with a central object server and several local object servers, you can store the documents in the local servers for quick access. After some time, those documents can be migrated to the remote central object server (they are archived in the central site, and may be copied to ADSM-managed optical or tape libraries).

You can define this powerful function with the NT System Administration program as follows:

1. Install a new object server. If you want your documents to be migrated from one server to other, then you need two servers. Assume that you have an NT object server and you install a new AIX object server.
2. Add the new object server into the servers and workstations network tables.
3. Start the library server and your two SMS servers.
4. Start the System Administration program. See Figure 31 on page 42.

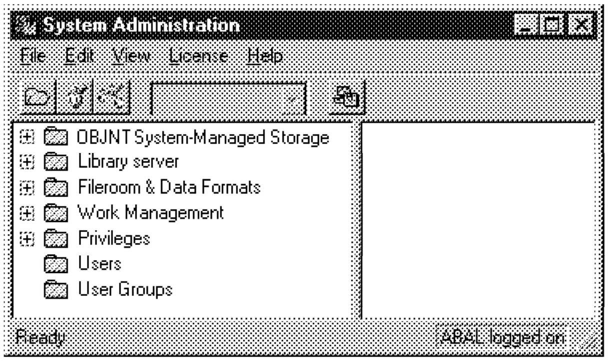
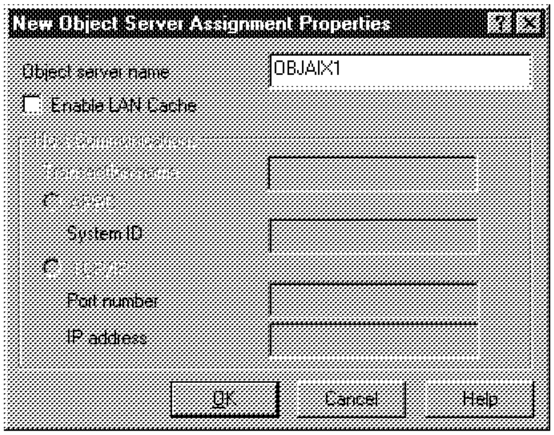


Figure 31. System Administration Application

5. Select **File, New, Library Server, Object Server Assignment**.



Object server name: OBJAIX1

□ Enable LAN Cache

Figure 32. Assign New Object Server

Write the name of your new AIX server and leave the Enable LAN Cache box blank. (This topic is discussed in 2.12, “LAN Cache” on page 45.) Now you have a new folder called System-Managed Storage, with two subfolders, one for each object server (see Figure 33).

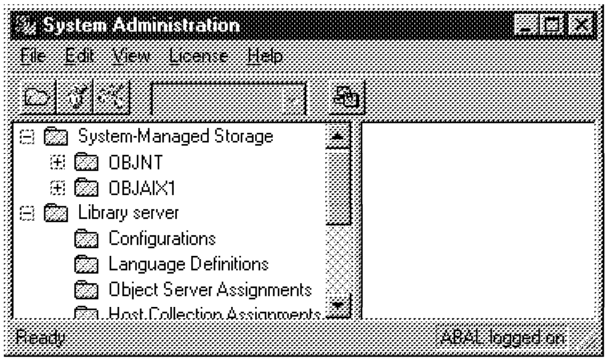
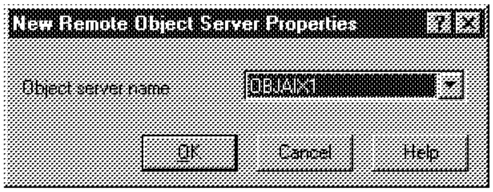


Figure 33. Administering Two Object Servers

6. Select the AIX object server and define all the SMS objects for the new AIX object server.

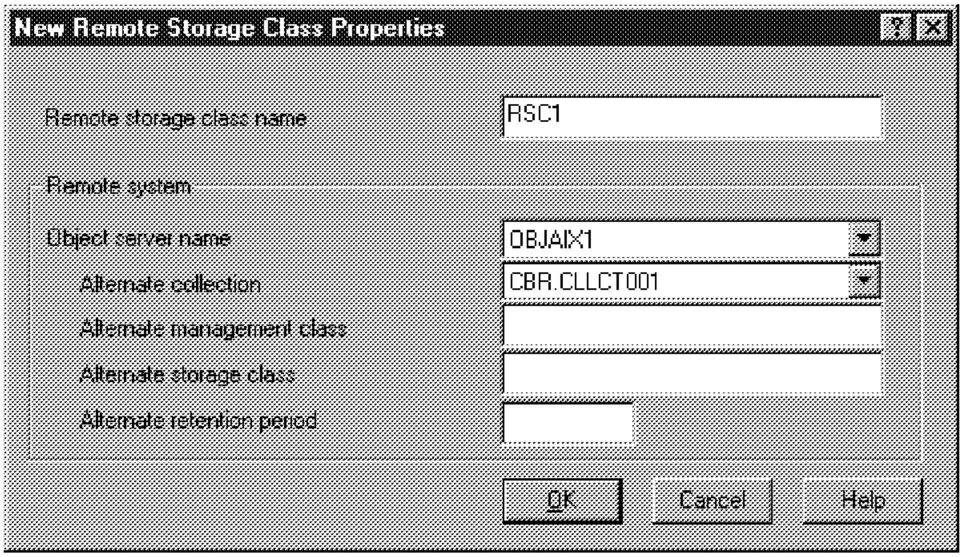
7. Select the NT object server and create a new remote object server (see Figure 34 on page 43).



Object server name: OBJAIX1

Figure 34. New Remote Object Server Definition

8. In the NT object server, create a new remote storage class (see Figure 35).



Remote storage class name: RSC1

Object server name: OBJAIX1

Alternate collection: CBR.CLLCT001

Alternate management class:

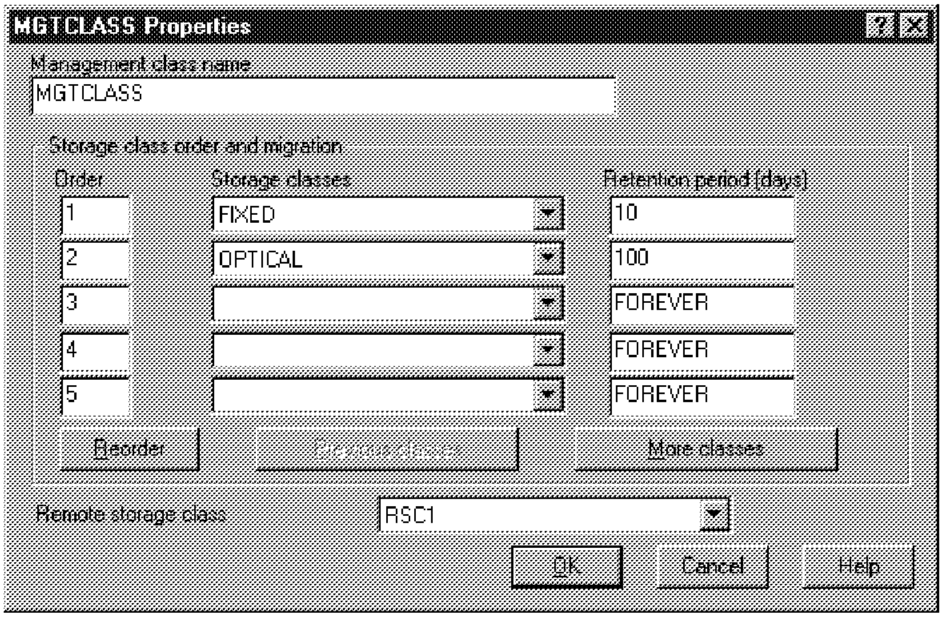
Alternate storage class:

Alternate retention period:

Figure 35. New Remote Storage Class Definition

In this window you write the name of the new remote storage class, select the remote object server, and select the collection that will be assigned to the documents in that remote server. You do not need to specify any other value; other values should not appear in that window.

9. Continue with the NT object server and open the properties windows of the management class you have to modify. Select the remote object class in the bottom window and click on **OK** (see Figure 36 on page 44).


MGTCLASS Properties

Management class name: MGTCLASS

Storage class order and migration

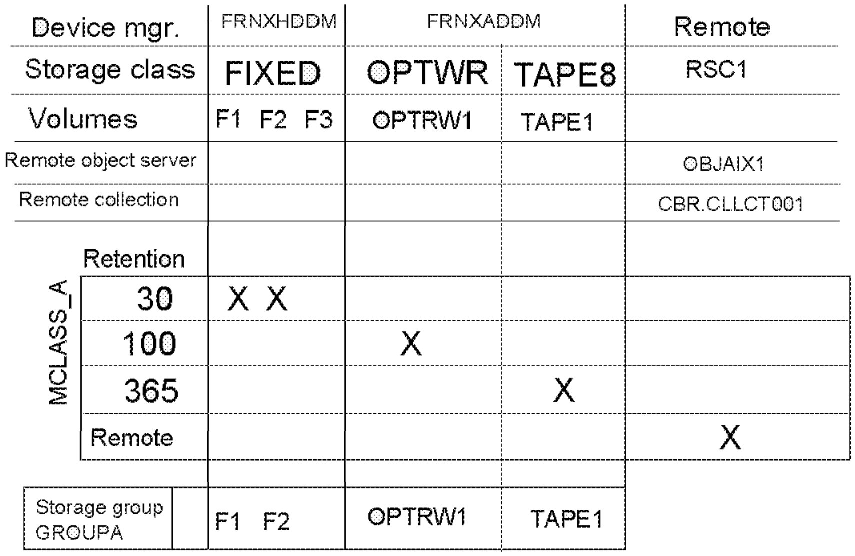
Order	Storage classes	Retention period [days]
1	FIXED	10
2	OPTICAL	100
3		FOREVER
4		FOREVER
5		FOREVER

Remote storage class: RSC1

Figure 36. Adding a Remote Storage Class to an Existing Management Class

That is all you have to do. Remember, however, that all those names and objects must be very carefully planned before you start configuring your system.

Figure 37 represents remote storage classes in our SMS charts.



Device mgr.	FRNXHDDM	FRNXADDM		Remote
Storage class	FIXED	OPTWR	TAPE8	RSC1
Volumes	F1 F2 F3	OPTRW1	TAPE1	
Remote object server				OBJAIX1
Remote collection				CBR.CLLCT001
MCLASS_A Retention				
30	X X			
100		X		
365			X	
Remote				X
Storage group GROUPA	F1 F2	OPTRW1	TAPE1	

Figure 37. SMS Chart Representation of a Remote Storage Class

2.12 LAN Cache

Some large VisualInfo systems are composed of a central library and object servers, which have to support many users located in several branch offices. Usually the communications lines between the central site and the remote users do not have a high bandwidth.

When a user requires a document, it is sent from the central object server to his workstation through a low bandwidth communications line. If that document is requested by another user in the same location, it is sent again from the central server to the workstation of that new user. If, the first time the document is transmitted, you can store it in temporary local storage or cache, then when another user needs to view it he can get it from that cache, avoiding transmitting the document again from the central repository. This approach improves the response time and optimizes the use of communication lines.

As shown in Figure 38, you can provide that temporary storage or cache by installing a VisualInfo object server near the end user and configuring that server as a LAN cache object server.

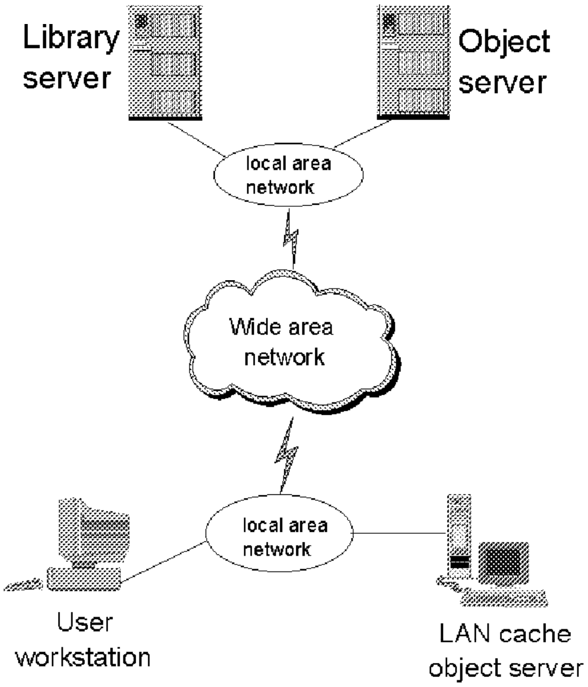


Figure 38. LAN Cache System

LAN cache works as follows:

1. The user sends a request to the library server to view a document
2. The library server, who knows that the user has a local cache defined, asks the local object server if it has the latest copy of that document.
3. If the latest version of the document exists in the DASD of the local object server, it is sent from the local object server to the user workstation. Otherwise, the library server tells the central object server to send the document to the user workstation.

4. At the same time, the library server tells the local object server to get the document from the central object server.

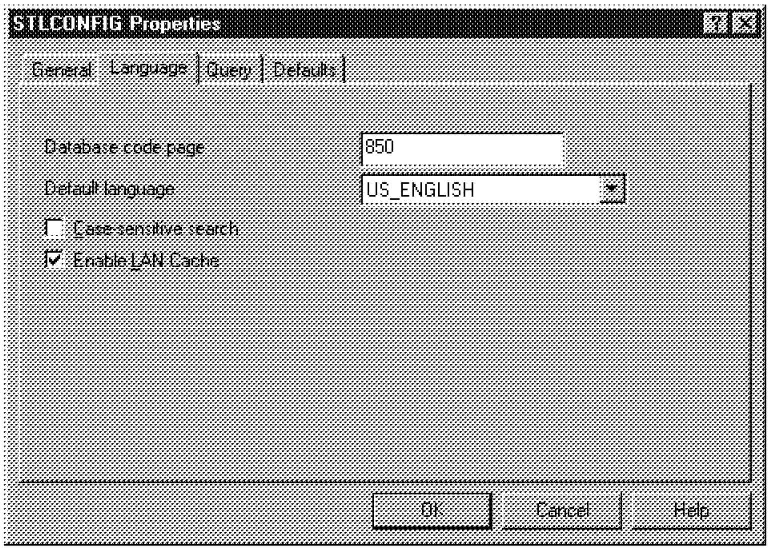
5. The local object server gets the document from the central object server and stores it in its DASD.

Important: The LAN cache is useful if the document is accessed more than two times and only viewed, not modified. Remember that the document is sent twice through the slow wide area network. If the document is then accessed only once, you overload the communication lines, and should turn off LAN cache.

To configure the LAN cache you have to do the following:

1. Enable the LAN cache in the library server configuration (see Figure 39).

   Note: You cannot modify the standard IBMCONFIG file. You must create a new config file with your desired changes. Then modify the startup of the library server (startup properties on NT and shell file on AIX) to point to the new config file.



Database code page: 850

Default language: US_ENGLISH

□ Case sensitive search

☑ Enable LAN Cache

Figure 39. LAN Cache System

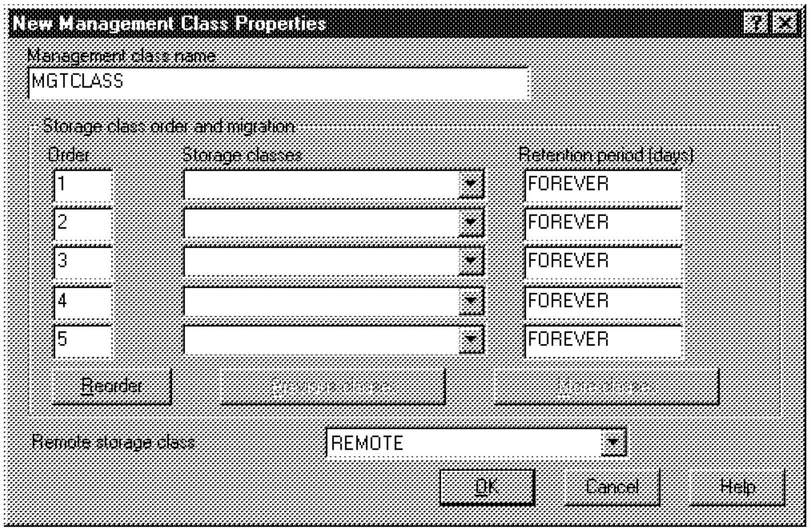
2. Check the Enable LAN Cache box when defining the new local object server (see Figure 32 on page 42). You have now enabled LAN cache.

The LAN cache object server stores the objects in the staging area.

Some users would like to have a remote object server, with LAN cache enabled, for quick response time, but do not want to back up this server. This can be done with standard VisualInfo definitions (though they go against the normal “rules” of VisualInfo configuring). Here is how this would be done.

1. Create a new management class in the local object server without a local storage class, using only the remote storage class, as shown in Figure 40 on page 47.

   This works because when the destager runs, it will have no storage class to move the objects to, and they will remain staged.


New Management Class Properties
Storage class order and migration
Reorder
Previous classes
More classes
OK
Cancel
Help

Management class name: MGTCLASS

Order	Storage classes	Retention period (days)
1		FOREVER
2		FOREVER
3		FOREVER
4		FOREVER
5		FOREVER

Remote storage class: REMOTE

Figure 40. LAN Cache Management Class Definition

2. Assign the new management class to a collection. When the destager and the migrator run, the objects will be copied to the new storage location. The copies will be left on the remote object server until they are eventually edged out by newly staged objects.

   The remote server does not have to be backed up, since the only exposed window is the amount of time before the next destager and migrator runs. If a failure occurs, the documents from this time period can be rescanned and reindexed.

3. Assign the local object server to the user as the default object server (see Figure 27 on page 37 and Figure 28 on page 37).

Chapter 3. VisualInfo and the ADSM Interface

This chapter describes how VisualInfo uses ADSM to store and retrieve objects on an optical library, the related ADSM APIs used for these functions, and environment variables to set in VisualInfo for this purpose. We assume reader familiarity with ADSM and VisualInfo.

3.1 VisualInfo and ADSM for Storage Management: the Interface

ADSM provides an application programming interface (API) that can be used to integrate a storage management solution with existing VisualInfo applications, to ensure that critical data is protected and easily recoverable. When VisualInfo uses the ADSM API, it becomes an ADSM application client that can communicate with an ADSM server to store or

To store and retrieve the objects for VisualInfo, the application client for an ADSM server (the VisualInfo object server) provides these services through the interface.

After an ADSM application client is installed, the user must modify the ADSM client options file to indicate the node name of the workstation and the method to use to communicate with the ADSM server.

The following points explain the VisualInfo and ADSM interface.

1. The interface begins with the installation of an ADSM client on the VisualInfo object server platform.
2. The client option file is then updated to indicate the client name, which should be the node name (hostname) of the platform where it is installed.
3. The environment variables of the object server user for using ADSM APIs must be set.
4. ADSM objects must be defined on the ADSM server, such as *Policy Domain, Policy Set, Management Class, Copy Group, Storage Pool and Volume* for the ADSM client node.
5. The client node (where the ADSM client is installed) must be registered in the ADSM *Policy Domain.* This lets the ADSM server know about the client in order to process its requests.
6. The object server is now ready to use ADSM APIs. When it does so, it is treated like an ADSM client by the ADSM server.
7. The object server can use ADSM APIs only when the device manager FRNXADDM for ADSM in VisualInfo system administration SMS components is enabled.

3.1.1 Environment Variables for the Object Server

It is the object server (and VisualInfo subcomponents) which interfaces with ADSM. The environment variables indicate the ADSM API locations to look for, when there is a need to use them.

Following are the environment variables which are to be set for the object server environment. Paths for each variable are detailed in the following chapters.

- DSMI_CONFIG
- DSMI_DIR
- DSMI_LOG

3.1.2 VisualInfo Functions and Related ADSM APIs

There are two object server components (migrator and destager) and a function for object retrieval which completes their functions by calling ADSM APIs. In the following sections, we discuss the process flow of each component and function.

3.1.2.1 Destaging Objects to ADSM

To destage an object, the destager reads the BASE_OBJECTS table to find the object attributes for the VisualInfo management class and storage class for the index class associated with that object. If a VisualInfo volume in that storage class is defined to use an ADSM management class, then the destager uses the following ADSM APIs to destage the object to the ADSM management class.

If the ADSM device manager is not enabled, then there is no way for any component of the object server to use ADSM APIs for that storage class and the associated management class. This is the main access barrier for the VisualInfo and ADSM interface that VisualInfo components have to cross to use ADSM APIs.

The Destager calls the following ADSM APIs to destage an object.

- dsmInit
- dsmQuerySessInfo
- dsmBeginQuery
- dsmGetNextQObj
- dsmEndQuery
- dsmTerminate

- dsmInit
- dsmQuerySessInfo
- dsmBeginQuery
- dsmGetNextQObj
- dsmEndQuery
- dsmTerminate

- dsmInit
- dsmBeginTxn
- dsmBindMC
- dsmSendObj
- dsmSendData
- dsmEndSendData
- dsmEndTxn
- dsmUpdateFS

- dsmTerminate

3.1.2.2 Migrating Objects to ADSM

The process flow for migration is similar to destaging. The migrator calls the following ADSM APIs to migrate an object.

- dsmInit
- dsmQuerySessInfo
- dsmBeginQuery
- dsmGetNextQObj
- dsmEndQuery
- dsmBeginQuery
- dsmGetNextQObj
- dsmEndQuery
- dsmTerminate

- dsmInit
- dsmBeginGetData
- dsmGetObj
- dsmGetData
- dsmEndGetObj
- dsmEndGetData
- dsmTerminate

- dsmInit
- dsmBeginTxn
- dsmBindMC
- dsmSendObj
- dsmSendData
- dsmEndSendObj
- dsmEndTxn
- dsmUpdateFS
- dsmTerminate

3.1.2.3 Retrieving an Object from ADSM

When the request for an object is sent by the client, the object server (after receiving the request from the library server) searches the BASE_OBJECTS table for the associated management class and the location of the object. It then calls the following ADSM APIs, if the object is in ADSM storage.

- dsmInit
- dsmQuerySessInfo
- dsmTerminate

- dsmInit
- dsmBeginQuery
- dsmGetNextObj
- dsmEndQuery
- dsmBeginQuery
- dsmGetNextObj
- dsmEndQuery
- dsmBeginGetData
- dsmGetObj
- dsmGetData
- dsmEndGetObj
- dsmEndGetData
- dsmEndGetObj
- dsmTerminate

For the description of what each API does, refer to Appendix C, “ADSM APIs Used by VisualInfo” on page 201. For a detailed description of these APIs, refer to *ADSTAR Distributed Storage Manager: Using the Application Programming Interface*.

Chapter 4. Two Example Configurations

These examples describe customer requirements for a document management solution. We explain the customer environment, the requirements for the solution, and an architecture that could be set up to meet the requirements.

We explain in detail which SMS components to define, as well as the components of prerequisite subsystems such as ADSM.

Remember that these are examples only. A complete definition of all the ADSM parameters is beyond the scope of this book, but you must define them with an ADSM specialist in a real system. The same goes for the database: in a real system you must define some parameters to get the best performance, and you will need the help of a DB2 specialist to define them properly.

4.1 Example 1: Windows NT

The customer in Example 1 is a medium-size bank looking for a system to manage its mortgage documents. The bank wants to start in its main branch office with a small system that eventually can be migrated to their S/390 mainframe.

In this first project the bank will have 30 users working on the system. The users have Windows-based workstations, and the departmental servers are Windows NT. The communication protocol used on the Token-Ring LAN is TCP/IP.

These 30 users process about 20 mortgages per day. They group all documents about each mortgage in a folder. Each folder has about 10 documents with 2 pages per document. Some of these documents come from word processors, others from e-mail. But most of them are paper documents that must be scanned and stored. One of these documents is a legal document that may contain more than 50 pages.

The bank has a well-defined business process, and the time between when the client asks for the service and the mortgage is granted or denied is never more than 90 days. During this time quick access to the documents is essential. It should take no more than 5 seconds to see the first page. When the users close the folder, all documents can be archived in a slower device. This migration must be done manually by the user, not on a time-scheduled basis.

The legal documents represent a special case. The users want to save all of those documents, but they know that the time to retrieve them could be excessive. So they decide to create two different documents: one with the first three pages (90% of the time they need only those pages) and one with all pages. The document with all pages can be archived directly into a low-cost storage device.

When a folder is closed, it must be stored for as long as 20 years, until the mortgage ends. Then it must be stored for another five years for legal reasons.

Sometimes a folder is required in the next 10 days after it is closed. So they decide to maintain the closed folder for 10 more days in fast storage before the system migrates it.

Finally, the bank does not want to use the VisualInfo client application. It wants to develop an application integrated with its Lotus Notes workgroup environment.

4.1.1 Analyzing the Storage Requirements

The first thing to do is to determine the space requirements for the storage system. The best way to make such a determination is to take some real documents from the customer, scan them and find an average size for each kind of document. Suppose that each document has an average size of 100KB, so each folder is 1 MB in size. We need space for 90 days of work; if 20 folders are processed each day we need 20*90*1 MB = 1.8GB of storage. Then we will define one volume of this capacity in DASD. Magnetic disks are required to achieve the 5-second maximum response time.

Every folder has a large document, about 50 pages in size, so about 2.5MB are required per document. Those documents are stored directly to optical disks.

The bank will need about 16GB per year of online long-term storage, and the best choice is an optical library. So the documents are migrated from DASD to optical disks. Some documents must be stored for more than 20 years, and the reliability of optical disks helps guarantee that information will not be lost.

4.1.2 Defining the Migration

One of the most important requirements is that the bank wants to migrate the documents when the folder is closed, not after waiting for a number of predefined days after being scanned. This is not a standard feature of VisualInfo, but we can use one of the SMS APIs to meet this requirement.

Note: A similar result could be obtained by reindexing into an index class associated with collection A.

We can define two management classes with these properties:

1. MGTCLASB
   - Storage class FIXED with retention period of 90 days
2. MGTCLASA
   - Storage class FIXED with retention period of 10 days
   - Storage class OPTICAL with retention period of 365*20 days

When the documents are indexed, they are assigned to MGTCLASB. We have defined a retention period of 90 days because all of the documents are processed in less than 90 days. If some documents are not processed in 90 days, they will remain on magnetic disk because there is no other storage class defined for migration.

When a user closes a folder, we change the management class of all its documents to MGTCLASA. The documents will stay on DASD for 10 days and then are moved to optical disk for 20 years. You can define a retention period of FOREVER, if you are sure that you will never need to migrate the documents to a third storage class.

The change in the management class can be made with the SimLibChangeObjectSMS API. See Appendix A, “Updating SMS Parameters through APIs” on page 189, for the use of this API.

Note: A similar result is obtained by reindexing into an index class associated with collection B.

For those big documents that are seldom accessed, we define a management class with storage class OPTICAL and a retention period of 365*20 days, so those documents are stored directly into the optical library. We will use Write Once Read Many (WORM) optical disks to store the documents because we will not have to modify them, and WORM disks are less expensive than R/W disks.

Note: Be careful when destaging directly to optical. This can impact your performance and use more system resources. See B.1.5, “Object Server Performance Tips” on page 194.

Figure 41, Figure 42 on page 56 and Figure 43 on page 56 show charts of the SMS objects. First, look at collection B.

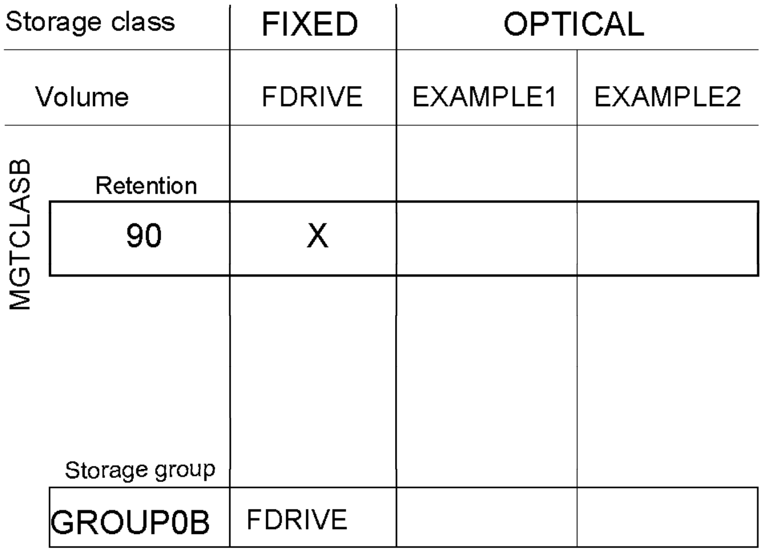
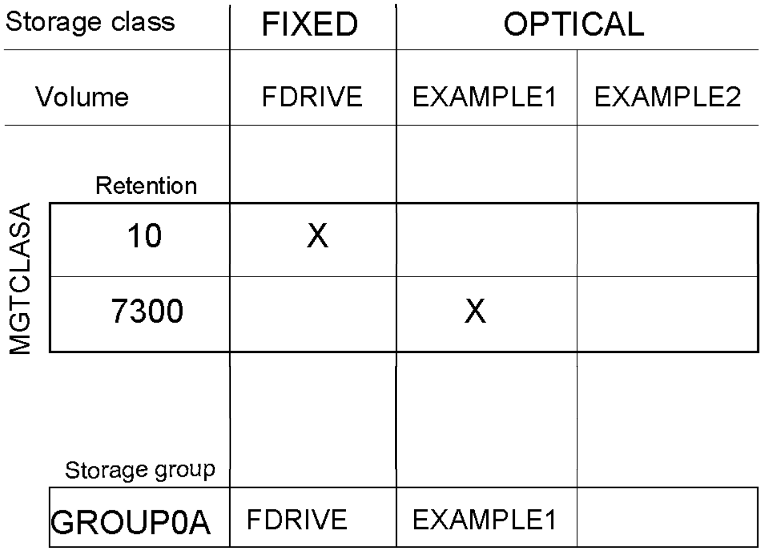


Figure 41. Properties of Collection CBR.CLLCTB

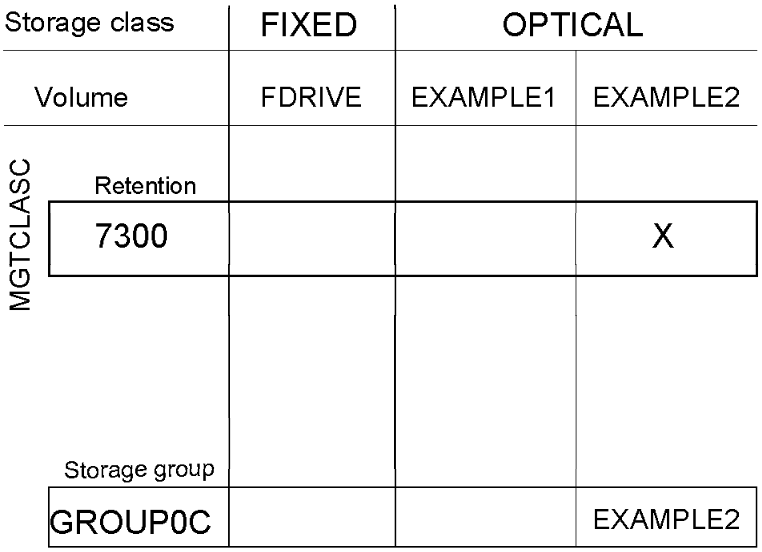
Next, look at collection A.



Storage class		FIXED	OPTICAL	
Volume		FDRIVE	EXAMPLE1	EXAMPLE2
MGTCLASA	Retention			
	10	X		
	7300		X	
	Storage group			
	GROUP0A	FDRIVE	EXAMPLE1	

Figure 42. Properties of Collection CBR.CLLCTA

Last, look at collection C.



Storage class		FIXED	OPTICAL	
Volume		FDRIVE	EXAMPLE1	EXAMPLE2
MGTCLASC	Retention			
	7300			X
	Storage group			
	GROUP0C			EXAMPLE2

Figure 43. Properties of Collection CBR.CLLCTC

To configure this example in our VisualInfo system, we have to create ADSM objects and VisualInfo SMS objects. See 5.8, "Configuring Example 1" on page 76, for a detailed description of this configuration.

4.2 Example 2: AIX

This example describes customer requirements for an image solution. We explain the environment in which the customer works, its image requirements, and a system architecture that could be set up to meet its requirement.

Then we explain in detail the System Managed Storage (SMS) and prerequisite subsystem (ADSM and VisualInfo Fileroom) components that should be defined to support the customer′s requirements.

4.2.1 Customer Environment

This customer is a government contractor whose objective is to deliver unemployment benefits and family allowances to selected people according to their family status and incomes (*clients*).

To manage its clients all around the country, this contractor has branch offices located in the major cities. Each branch office should be provided with an image solution.

4.2.2 Customer Requirements

Every day the branch offices receive several thousand letters from their clients. The customer′s requirements for an image solution that can handle this amount of documents are:

1. Scan received letters in batch mode.
2. Index documents in batch mode.
3. Store documents on quick access media for some weeks.
4. Migrate documents from quick access media to slow access media.
5. Gather documents with the same life span on the same storage media.
6. Ensure document availability.

4.2.3 System Architecture

The system architecture shown in Figure 44 on page 58 is designed to support the customer′s requirements.

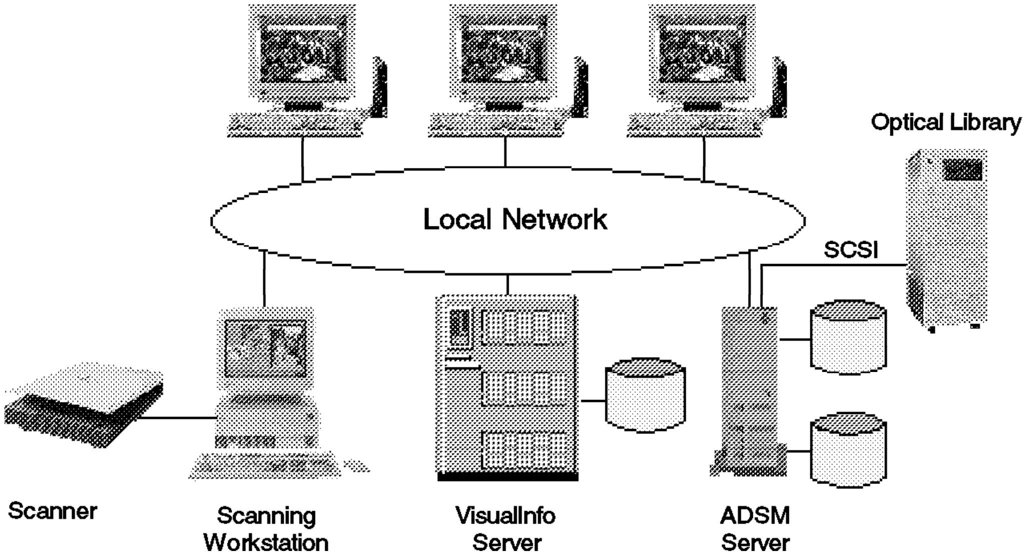


Figure 44. System Architecture Configuration

It includes the following components:

- A scanning workstation

  A workstation connected to an advanced scanner is used to scan batch documents (approximately 100 documents per batch). It runs a scanner application.

  Bar codes are used to identify the documents.

  Every night a scheduler is used to start the indexing of all documents stored in a VisualInfo workbasket.

  A folder manager API is used in this workstation to:

  – Store scanned documents in a VisualInfo server workbasket.

  – Index and store the document in the VisualInfo server.

- VisualInfo server

  The VisualInfo server includes the VisualInfo library and object servers. It keeps track of the indexes and stores documents, so it can process storage and retrieval requests.

- ADSM server

  An ADSM server is used to manage the storage of documents in an optical library. It is not necessary to have a dedicated ADSM server for VisualInfo. If you already have the fully licensed ADSM code running on another server, you can use that server for your VisualInfo system as well.

- User workstations

  User workstations run programs that use the folder manager API to retrieve documents from the VisualInfo server based on their index class information.

- VisualInfo system administration program

  One of the workstations runs the VisualInfo system administration program used to set up and administer the VisualInfo server.

4.2.4 ADSM, VisualInfo Fileroom and SMS Component Definitions

Based on this customer′s system architecture, we set up an environment (see Chapter 6, “Setting Up Example 2 on AIX” on page 85) to configure the necessary SMS resources and their prerequisite components (ADSM and VisualInfo filerooms).

In this book we do not cover the scanning and retrieving of documents or the performance issues in this environment.

Data availability is covered in Chapter 7, “Backing Up and Restoring VisualInfo Data” on page 163.

For sizing, we make the following assumptions:

- The average number of letters received by a branch office is 2000.
- The average size of a letter (scanned document) is 100KB.

The following sections define the resources in ADSM and VisualInfo SMS to comply with the customer′s requirements.

4.2.4.1 ADSM Components

Documents with the same life span should be gathered on the same optical library platters (or volumes) when migrated. Therefore, we define two storage pools in ADSM:

- SPE2ST (short-term storage pool for documents with a two-year lifetime)
- SPE2LT (long-term storage pool for documents with a five-year lifetime)

Each storage pool contains volumes (optical library platters). In our example, we define the following volumes:

- VI001A in storage pool SPE2ST
- VI002A in storage pool SPE2LT

Two more volumes, VI003A and VI004A, are defined as scratch volumes.

For each storage pool, a management class containing a backup copy group should be defined. Therefore, we create two management classes:

- MCE2ST (short-term management class for documents with a two-year lifetime)
- MCE2LT (long-term management class for documents with a five-year lifetime)

Then we have to define the backup copy group to use in each management class and associate it with the related storage pool.

The management classes must be stored in a policy set, so we create a new policy set PSE2, and include it in the policy domain STANDARD.

4.2.4.2 VisualInfo Fileroom

When a document is indexed, it is recorded in a VisualInfo database and then stored and migrated according to the SMS collection to which it belongs.

The collection to which a document belongs is specified in the index class used for indexing a document. For our configuration, we have defined two index classes:

- Index class for documents with a two-year lifetime --> Document for short-term
- Index class for documents with a five-year lifetime --> Document for long-term

4.2.4.3 SMS Components

When indexing a document, we create two collections in SMS to ensure that the documents are stored on specific volumes according to their required lifetimes:

- CNE2ST (short-term collection for documents with a two-year lifetime) documents)
- MCE2LT (long-term collection for documents with a five-year lifetime)

Each collection contains a management class and a storage group, so we create two management classes and two storage groups:

- MCVIST (short-term management class for documents with a two-year lifetime)
- MCVILT (long-term management class for documents with a five-year lifetime)
- SGST (short-term storage group)
- SGLT (long-term storage group)

The customer wants to store all documents first on quick access media (DASD) and then migrate them to long-term storage, so we have to create two storage classes in each management class:

- SCDAST (storage class using DASD and recorded in management class MCVIST)
- SCOPST (storage class using optical and recorded in management class MCVIST)
- SCDALT (storage class using DASD and recorded in management class MCVILT)
- SCOPLT (storage class using optical and recorded in management class MCVILT)

  Note: In this example, we create two storage classes for storing documents on DASD. For this configuration, it should also be possible to create only one storage class called DASD and have it used by two different collections.

To test our configuration, we define a retention period of 1 day on DASD and 100 days on optical.

Then we must define, by storage group, which volume to use. The volume name is generated by SMS.

A DASD volume name is that of the related file system.

An optical volume name is the concatenation of "ADSM:" with the name of the ADSM management class used. For our configuration the following VisualInfo volumes are defined:

- /SHORTTERM (volume for DASD and included in storage class SCDAST)
- ADSM: MCE2ST (volume for optical and included in storage class SCDAST)
- /LONGTERM (volume for DASD and included in storage class SCDALT)
- ADSM: MCE2LT (volume for optical and included in storage class SCDALT)

Every day, 2000 documents are scanned. Therefore at least 200 MB should be available in the staging area. To cope with two and one-half days of scanning work, we define a staging area of 500 MB. Table 1 summarizes the different resources needed to set up Example 2.

Table 1. Resources to Define for Example 2

SMS Resources	
Collection	CNST, CNLT
Management class	MCVIST, MCVILT
Storage group	SGST, SGLT
Storage class	SCDAST, SCOPST, SCDAST, SCOPLT
Volume	/SHORTTERM, ADSM: MCE2ST, /LONGTERM, ADSM: MCE2LT
ADSM Resources	
Storage pool	SPE2ST, SPE2LT
Volume	VI001A, VI002A, VI003A, VI004A
Management class	MCE2ST, MCE2LT
Policy set	PSE2
VisualInfo Resources	
Index class	Document for short-term, document for long-term

Chapter 5. Setting Up Example 1 on Windows NT

In this chapter we install and configure the software needed to implement a complete VisualInfo system on a Windows NT platform with an ADSM-managed optical library. Then we configure example 1 (see 4.1, “Example 1: Windows NT” on page 53 for a complete description). We do not explain in detail the installation of VisualInfo, DB2, and VisualC++, but we discuss some points to consider in installing those products.

5.1 Configuration for Example 1

In Example 1, we use one computer for three servers: the VisualInfo library server, the VisualInfo object server, and the ADSM server (see Figure 45). There are no differences in the SMS and ADSM installation and configuration if you use one computer for the VisualInfo Library Server and another for the VisualInfo Object Server and the ADSM server. But remember that the library server is the component that requires more resources, and is good practice to put it on a separate computer. The object server and ADSM server require fewer resources and can share the same PC. Figure 45 shows this configuration.

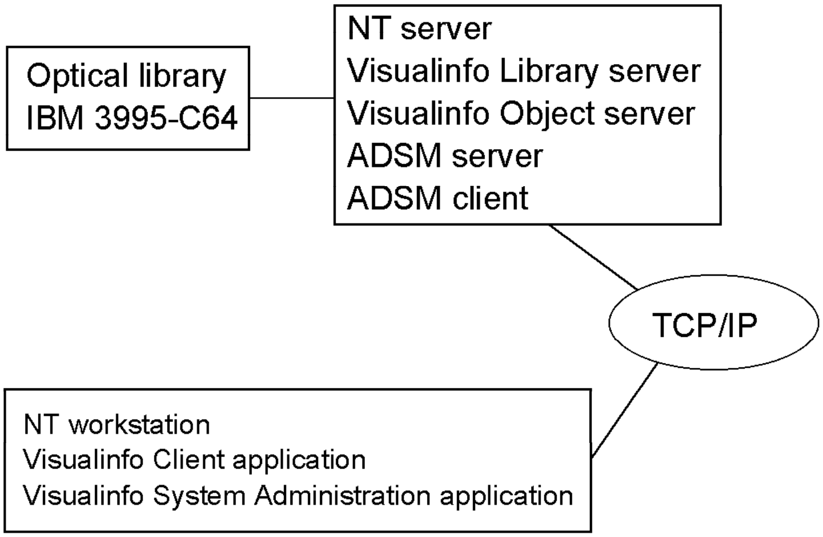


Figure 45. Setup for Example 1

To import documents into the system and to configure SMS, we use a workstation on Windows NT.

5.2 Example 1 Hardware and Software

The computers used for example 1 are:

- An IBM PC750, with a Pentium 120MH CPU, 80 MB of memory, and two disks of 1GB for the servers
- An IBM PC750, with a Pentium 120MH CPU, 48 MB of memory, and two disks of 1GB for the VisualInfo client workstation.

The optical library is an IBM 3995-C64 connected to an Adaptec SCSI AHA-2944 PCI differential adapter card located in the server PC. This library has two drives and space for 104 optical disks.

The computers are connected to a 16Mbps Token-Ring network.

It is very important to have enough memory in the server. The VisualInfo guide recommends 96 MB when the library and the object servers share the same machine, and a physical paging space on hard disk large enough that the sum of the physical paging space and RAM is at least 300 MB.

Every VisualInfo system has its own requirements, based on such factors as the number of concurrent users and the number of documents scanned per day,and you have to configure your hardware to satisfy those requirements. See *Analyzing VisualInfo Performance and Capacity*, SG24-4974 for more information.

This is the software we used for Example 1.

- Windows NT Server V4 Service Pack 3
- DB2 UDB V5
- ADSM V3.1.0 + Service Pack 1
- Microsoft VisualC++ V5
- VisualInfo V2.3.1

5.3 Installing Windows NT Server V4

Install the Windows NT server according to your company rules. We recommend installing the system in a FAT partition. This will allow you to recover easily if needed, by booting with a DOS diskette and making repairs on the system.

We recommend that you create a new user with administrative privileges, and log on to the system with that user. Also install all the software with that user. You can use any name but keep it short, no more than eight characters, because of the DB2 limitation on user names for compatibility with host systems.

You can also change the NT setting to log on automatically to that user every time the system starts.

To have a high-performance system, you can follow some guidelines when choosing the disks and partitions to create. See *Analyzing VisualInfo Performance and Capacity*, SG24-4974 for more information.

Verify the Installation

You now have a Windows NT system running. Verify that TCP/IP is working before proceeding with the next step.

5.4 Installing ADSM Version 3.1 Components

Installing the ADSM server for NT is simple:

1. Insert the ADSM for Windows NT CD into your CD-ROM drive (the ADSM CDs are provided with the VisualInfo box). The setup program will run when you load the CD-ROM into your drive. Otherwise, run the SETUP.BAT program in the root directory of the CD-ROM.
2. Select **Typical install.**
3. Select the default option: **Put the admin and backup clients in separate directories.**
4. Select the directory location where ADSM will be installed, C:WIN32APPIBMADSM by default.
5. Reboot the machine after the installation is completed.
6. Get the ADSM V3.1 PTF1 for NT server from the ADSM Web site, `http://www.storage.ibm.com/software/adsm/adsercli.htm`. Install the PTF, following the instructions in the v3ntptf1.txt file.

From this installation, you have an initial ADSM server configured. You have the default administrative client and the default backup archive client registered (see Table 2).

Table 2. Clients Registered during Installation of ADSM Server for NT.

Client	ID Name	Password
Administrative client	admin	admin
Backup archive client	client	client

5.4.1 Verifying the Installation

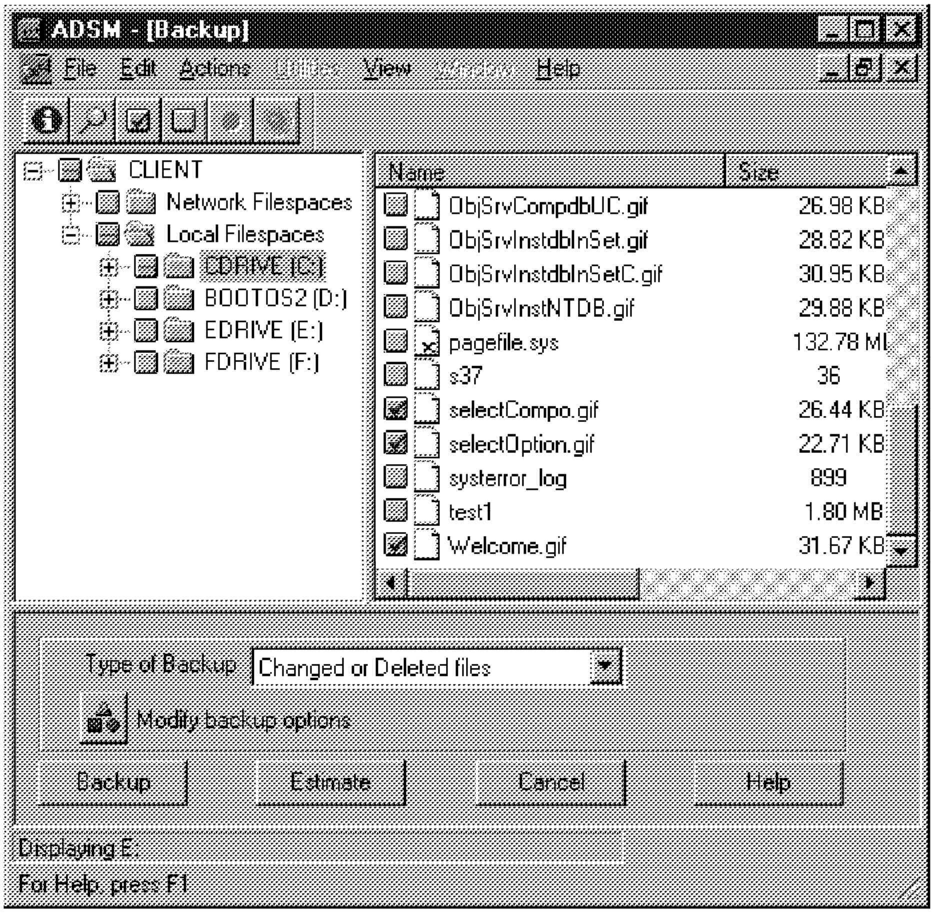
To verify that the components have been installed correctly, use the default client to make a backup and restore of some files:

1. Start the ADSM server.
2. Start the ADSM backup client and log on with node name CLIENT. You see this message in the ADSM server window:

```
adsm>
ANR2862W Session 1 for client CLIENT is using a network communications method
(Tcp/Ip). This server is NOT licensed for NETWORK connections.
ANR2841W Server is NOT IN COMPLIANCE with license terms.
ANR0406I Session 1 started for node CLIENT (WinNT) (Tcp/Ip 127.0.0.1(1032)).
```

   Do not worry about this message. You will change the communication protocol after this verification.

3. After logging on, click on **Backup**.
4. You will see a list of files, as in Figure 46 on page 66. Select some files and click on **Backup**.



	Name	Size
□	ObjSrvCompdbUC.gif	26.98 KB
□	ObjSrvInstdbInSet.gif	28.82 KB
□	ObjSrvInstdbInSetC.gif	30.95 KB
□	ObjSrvInstNTDB.gif	29.88 KB
□	pagefile.sys	132.78 M
□	s37	36
☑	selectCompo.gif	26.44 KB
☑	selectOption.gif	22.71 KB
□	systerror_log	899
□	test1	1.80 MB
☑	Welcome.gif	31.67 KB

Type of Backup: Changed or Deleted files

Displaying E:

For Help, press F1

Figure 46. ADSM Backup Client

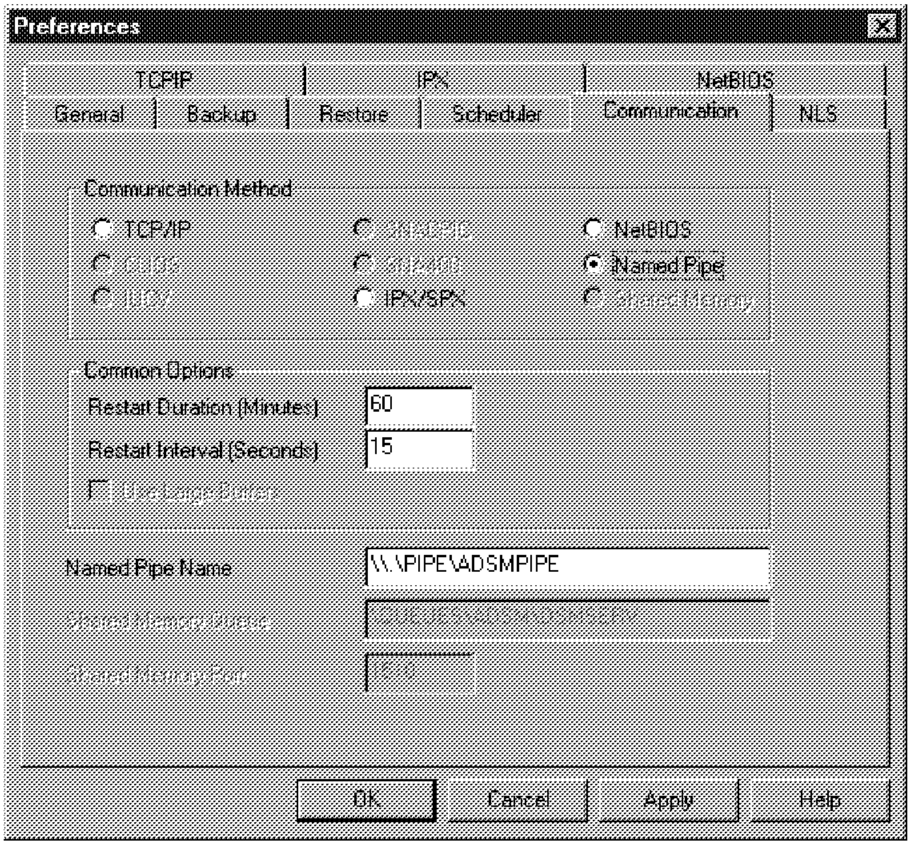
5. The files selected are copied into the ADSM system.

5.4.2 Configuring the Communication Protocol

Now you have the ADSM system running. By default the ADSM server listens for incoming client connections on TCP/IP and named pipes, and the client uses TCP/IP to connect to the server. The test you have just done has used TCP/IP because this is the default protocol for the client.

Since the ADSM client and the ADSM server are in the same machine, you get better performance if you use *named pipes*. Follow these steps to change the communication method for the client:

1. Start the ADSM Backup Client application.
2. After logon, click on **Edit** and **Preferences.**
3. Click on the **Communication** tab.
4. In the Preferences window (Figure 47 on page 67), select **Named Pipe** as the communication method and leave the default value for the name of the named pipe.


Preferences

TCPIP | IPX | NetBIOS

General | Backup | Restore | Schedular | Communication | NLS

Communication Method

○ TCP/IP ○ SNACPIC ○ NetBIOS

○ CLIOS ○ SNA400 ● Named Pipe

○ IUCV ○ IPX/SPX ○ Shared Memory

Common Options

Restart Duration (Minutes) 60

Restart Interval (Seconds) 15

□ Use Large Buffer

Named Pipe Name \\.\PIPE\ADSMPIPE

Shared Memory Queue

Shared Memory Port

Figure 47. ADSM Backup Client Communication Method

5. Click on **OK** and close the application.

Back up and restore some files to be sure the communication protocol is working. Now you should see this message in the ADSM server window:

```
ANR0400I Session 2 started for node CLIENT <WinNT> <Named Pipe>.
```

5.5 Installing ADSM Support for the 3995 Optical Library

To use the optical library with ADSM, you must define the device and configure some parameters to Windows NT and to the ADSM server. Before doing these steps, be sure that the library is connected and ready to run.

Attention

The 3995 should be installed by qualified personnel who perform some basic tests to ensure it functions properly. Do not go further before they have done their job.

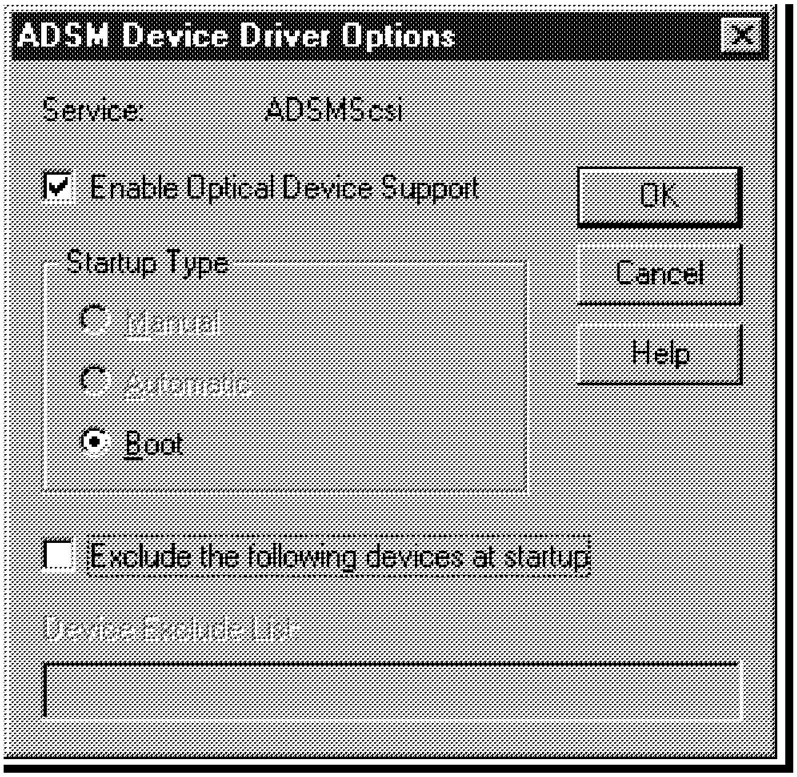
This section describes how to enable the optical support and define some ADSM objects. Then the optical library is tested to be sure it is working correctly with ADSM. Follow these steps:

1. Open the System Properties program in the Control Panel folder and make a copy of your current hardware profile.

2. Open the Devices program in the Control Panel folder and change the startup options for ADSMSCSI.SYS to BOOT.

3. Start the ADSM Server Utilities program, maximize the Services window, select the **Device Driver** tab, and click on **Options...**.

4. In the ADSM Device Driver Options windows (Figure 48) check the **Enable Optical Device Support** option and click on **OK**.



ADSM Device Driver Options

Service: ADSMScsi

☑ Enable Optical Device Support

Startup Type

○ Manual

○ Automatic

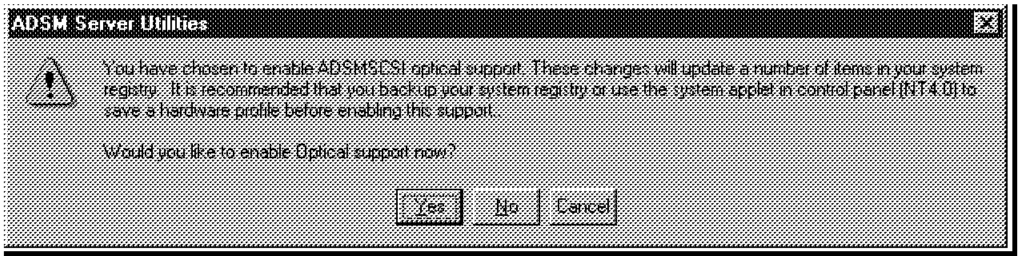
● Boot

□ Exclude the following devices at startup

Device Exclude List

Figure 48. ADSM Device Driver Options

5. Click on **Yes** in this warning message:



ADSM Server Utilities

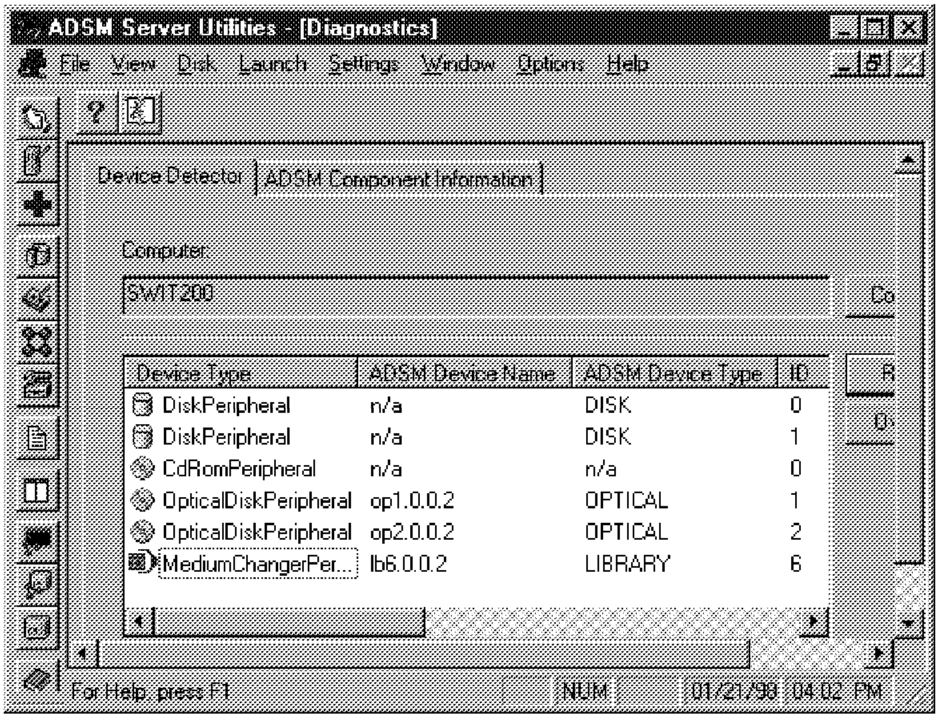
You have chosen to enable ADSMSCSI optical support. These changes will update a number of items in your system registry. It is recommended that you backup your system registry or use the system applet in control panel (NT4.0) to save a hardware profile before enabling this support.

Would you like to enable Optical support now?

Figure 49. Enabling Optical Support

Close all programs and restart your system. Now the optical drivers should not show up under the folder My Computer.

Start the ADSM Utilities program and maximize the Services window. The ADSMSCSI service should be running. Start the ADSM server service and maximize the Diagnostics window (Figure 50 on page 69).


ADSM Server Utilities - [Diagnostics]
File View Disk Launch Settings Window Options Help
Device Detector
ADSM Component Information
Co
R
O
For Help, press F1
NUM
01/21/98 04:02 PM

Computer: SWIT200

Device Type	ADSM Device Name	ADSM Device Type	ID
DiskPeripheral	n/a	DISK	0
DiskPeripheral	n/a	DISK	1
CdRomPeripheral	n/a	n/a	0
OpticalDiskPeripheral	op1.0.0.2	OPTICAL	1
OpticalDiskPeripheral	op2.0.0.2	OPTICAL	2
MediumChangerPer...	lb6.0.0.2	LIBRARY	6

Figure 50. ADSM Server Utilities - Diagnostics Window

Remember the parameters of the optical and library ADSM device types. You will need them later. Table 3 lists the types in our system:

Table 3. Information from the ADSM Device Detector

Device Type	ADSM Device Name	ADSM Device Type
OpticalDiskPeripheral	op1.0.0.2	OPTICAL
OpticalDiskPeripheral	op2.0.0.2	OPTICAL
MediumChangerPeripheral	lb6.0.0.2	LIBRARY

6. Go to the ADSM for Windows NT folder and open the ADSM admin command line window (the user ID is admin) and enter these commands:

   - To manage the optical library the ADSM server must have the license for Advanced Device Support. Type:

     ```
     adsm> register license file(advdev.lic)
     ```

   - Define the optical library. In our system, we assigned it the name c64lib. You can choose a different name for your system. The lb6.0.0.2 parameter is from Table 3:

     ```
     adsm> define library c64lib libtype=scsi device=lb6.0.0.2
     ```

   - Define a new device class with name c64opt. The mountlimit parameter specifies the maximum number of files that can be simultaneously open for input/output. There are two optical drives in the library, so use a value of 2:

     ```
     adsm> define devclass c64opt devtype=optical library=c64lib
     mountlimit=2
     ```

   - Define the two optical drives. Use the arbitrary names drive1 and drive2. The other device parameters are from Table 3. The element parameter specifies the element address of the drive. You must use the values 1 and 2 for the 3995-C64.

```
adsm> define drive c64lib drive1 device=op1.0.0.2 element=1
adsm> define drive c64lib drive2 device=op2.0.0.2 element=2
```

- Define a new storage pool with the arbitrary name c64optpool and associated with the c64opt device class we defined before. The maxscratch parameter specifies the maximum number of scratch volumes that can be requested by ADSM. This value can be updated later, so use a value of 3 (for example).

  ```
  >define stgpool c64optpool c64opt maxscratch=3
  ```

7. For the next command, start an MSDOS window, go to the \WIN32APP\IBM\ADSM\UTILS directory, and enter:

   ```
   dsmlabel -drive=op1.0.0.2,1 -library=lb6.0.0.2  -overwrite  -keep
   ```

   You see this message:

   ```
   ADSTAR Distributed Storage Manager for Windows NT
   Volume Label Utility Program
   Version 3, Release 1, Level 0.1

   Licensed Materials - Property of IBM

   5639-C59 (C) Copyright IBM Corporation 1994, 1997.  All rights reserved.
   U.S. Government Users Restricted Rights - Use, duplication or disclosure
   restricted by GSA ADP Schedule Contract with IBM Corporation.

   ANR9739I Initializing library device 'LB6.0.0.2'.

   ANR9732I Insert a new volume in the entry/exit port of library 'LB6.0.0.2' with
   side A facing up, then enter the name (1-32 characters) to be used for its
   label; or just press ENTER to quit this program:
   ```

   This is an utility program to label volumes. Insert a new optical cartridge in the I/O port of the library and write the name you want to assign to this volume (SVT001 in this example):

   ```
   SVT001
   ANR9743I Attempting to label volume 'SVT001' using drive 'OP1.0.0.2'...
   ANR9720I Volume 'SVT001' was labeled successfully using drive 'OP1.0.0.2'.

   ANR9732I Insert a new volume in the entry/exit port of library 'LB6.0.0.2' with
   side A facing up, then enter the name (1-32 characters) to be used for its
   label; or just press ENTER to quit this program:
   C:\WIN32APP\IBM\ADSM\utils>
   ```

   With this utility you label the disks to be used in the library. The parameters used are explained as follows:

   - -drive=op1.0.0.2,1 is the device name of the optical drive we use to perform the operation.
   - -library=lb6.0.0.2 is the device name of the optical library.
   - -overwrite changes the label of already labeled volumes. If your cartridges are new, this parameter has no effect.
   - With -keep, the inserted volumes stay in the jukebox.

8. After labeling the volumes, you must inform the ADSM server which volumes are available in a library by using the CHECKIN LIBVOLUME command. Go to the ADSM Admin Command prompt and type:

```
adsm >checkin libvolume c64lib status=scratch search=yes
```

where c64lib is the name of the optical library we defined before; status=scratch is used to define the volumes as scratch volumes so they can be dynamically added to the storage pools that need a new volume; and search=yes tells ADSM to search the library for all volumes.

ADSM starts a background process. When the process ends, you can verify which cartridges have been checked in by issuing the QUERY LIBVOLUME command, which displays the ADSM inventory of the library (see Figure 51):

```
adsm> query libvolume c64lib

Library Name     Volume Name     Status               Last Use
------------     -----------     -----------------    ---------
C64LIB           SVT001          Scratch(Optical)
C64LIB           SVT002          Scratch(Optical)
C64LIB           SVT003          Scratch(Optical)
C64LIB           SVT004          Scratch(Optical)
```

Figure 51. Volumes Checked in to the Optical Library

9. To back up files directly to optical, you can redefine the BACKUPPOOL storage pool. Enter these commands:

   ```
   adsm> update stgpool backuppool nextstgpool=c64optpool
   adsm> update stgpool backuppool access=unavailable
   ```

   The standard BACKUPPOOL is not available, so the backups must be stored in the next storage pool, C64OPTPOOL.

10. Verify that the process has been successful. Start a backup client session and back up some files so that they are stored in the optical library.

5.6 Installing VisualInfo

Before installing VisualInfo, you must install DB2 and VisualC≈ . Follow these steps:

1. Install DB2 UDB from the CD-ROMs you get with your VisualInfo package. Select **DB2 UDB Workgroup Edition**. You can select a **typical install** (118.1 MB) or a **compact install** (45.5 MB). We recommend a typical install, because with it you get all the documentation and some useful tools to query and configure your system.

   When you are asked to enter the administrator name, use the name you created when installing Windows NT (see 5.3, “Installing Windows NT Server V4” on page 64). If you get a warning, do not worry; click on **OK** and continue with the installation.

2. After the installation is completed, do not restart the system. Insert the DB2 Software Developer′s Kit in the CD-ROM drive and install it. Select a typical install, which will use 50 MB of your disk.

3. After installing the DB2 Software Developer′s Kit, reboot the machine and run the DB2 First Steps program from the DB2 for Windows NT folder. Create the SAMPLE database to be sure that the installation was successful.

Note

The installation program leaves the Control Center utility in your startup folder. You do not have to start it every time the system initializes, so move it from that folder to the desktop.

4. Install VisualC++. Select **Typical Install** (136.6 MB). Follow the instructions in Chapter 6 of the *VisualInfo Planning and Installation Guide*, GC31-7772 for the settings of the VisualC++ environment variables. After completing the installation, create a sample program and verify that you can compile and run it.

5. Install the VisualInfo library server and object server (30.6 MB). You can install the client and the system administration applications if you want to use them on the server.

   Verify the installation by importing or scanning a document with the client application. If you do not have a scanner, you can import one of the TIFF files in the FRNROOT directory.

   Create at least one more user with MAXPRIVS privileges. Reserve the default FRNADMIN user for emergencies, or delete it for security reason. If your workstation hangs before logging off, the VisualInfo session will stay open, and you will have to log on with FRNADMIN (or a similar user ID which you create) to the system administration program, select **License** and **List Users**, highlight the user logged on, and click on **Log off...**.

   Create one new index class to verify that the compiler is working. When you create new index classes, VisualInfo uses the C compiler to create a new DLL. If you get some errors when creating the DLL, verify the compiler settings.

5.7 Testing the Object Server-ADSM Interface

Now you have all the programs and subsystems installed and verified. We strongly recommend that you make a backup copy of all your systems. After this backup, test that the interface between the object server and the ADSM server is working:

1. Start the ADSM Server.
2. Start the ADSM Administrative GUI client and add a new node.

   Important

   The ADSM licensing you get with VisualInfo allows only one client license, the one that object server uses. Be sure you have purchased another license before creating a new user.

   Double-click on **Nodes**, select **Edit, Add**, and fill in the required fields of the General, Password, and Filespace tabs:

   - Node Name = PEPE200
   - Policy Domain = STANDARD
   - Compression = Client Determined
   - Enter any password
   - Check **Can delete backup data**

Click on **Add** to create the new node.

3. Edit a new file with name dsm.opt in the c:\win32app\ibm\adsm\api\dll directory and write the following:

   ```
   PASSWORDACCESS GENERATE
   LANGUAGE AMENG
   COMMMETHOD NAMEDPIPE
   NAMEDPIPENAME \\.\PIPE\ADSMPIPE
   NODENAME PEPE200
   TAPEPROMPT NO
   INCLUDE ?:STANDARD* STANDARD
   ```

   This dsm.opt file is the options file that VisualInfo uses to access ADSM. The nodename must be the name of your system (PEPE200 in ours). The PASSWORDACCESS GENERATE parameter indicates that the password is saved by ADSM and the object server can log on automatically to ADSM. When the password expires, another one is automatically generated and saved.

   In the last line, the first field definition (STANDARD*) is the file qualifier; the second field definition (STANDARD) assigns the specified management class to all generated files.

4. Generate the ADSM client password. Open an MSDOS window and type:

   ```
   set dsm_config=c:\win32app\ibm\adsm\api\dll\dsm.opt
   ```

   to override your current backup client setting. Then log on as a standard backup command line client:

   ```
   cd c:\win32app\ibm\adsm\baclient
   dsmc
   ```

   Issue an ADSM client query to connect to the ADSM server. For example, enter:

   ```
   query mgmtclass
   ```

   and enter the password assigned in step 3. Exit the ADSM backup command line client by entering `quit`. Now the password is stored encrypted in your system, and the object server could start an ADSM session without prompting for the password.

5. Copy the DSCAMENG.TXT file from the win32app\ibm\adsm\saclient directory to win32app\ibm\adsm\api\bin.

6. Edit the FRNNSTAD.BAT file of the FRNROOT directory and modify, if necessary, the statement:

   ```
   SET ADSMPATH=d:\win32app\ibm\adsm\api
   ```

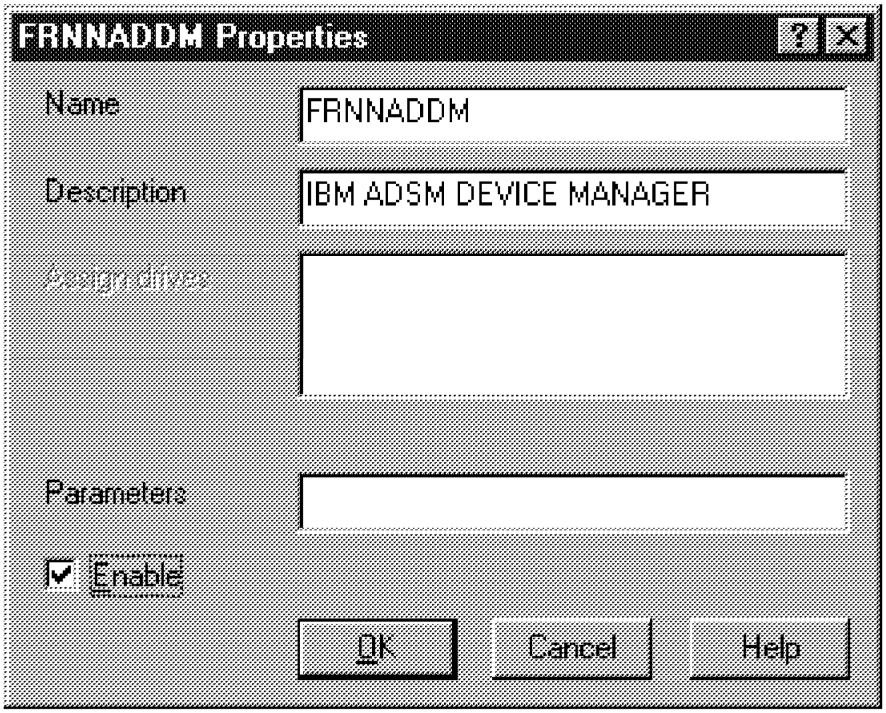
   to the correct drive and path where ADSM was installed. This variable is used to set other variables such as DSMI_CONFIG, DSMI_DIR and DSMI_LOG.

7. Start the VisualInfo library server.

8. Start the VisualInfo SMS server.

9. Start the VisualInfo System Administration utility.

10. Double-click on **OBJSRVRN System-Managed Storage**, double-click on **Device Managers**, open the FRNNADDM properties window (Figure 52 on page 74) and check the Enable box. Otherwise the ADSM device manager will not be available to the VisualInfo object server.



Name: FRNNADDM

Description: IBM ADSM DEVICE MANAGER

Parameters:

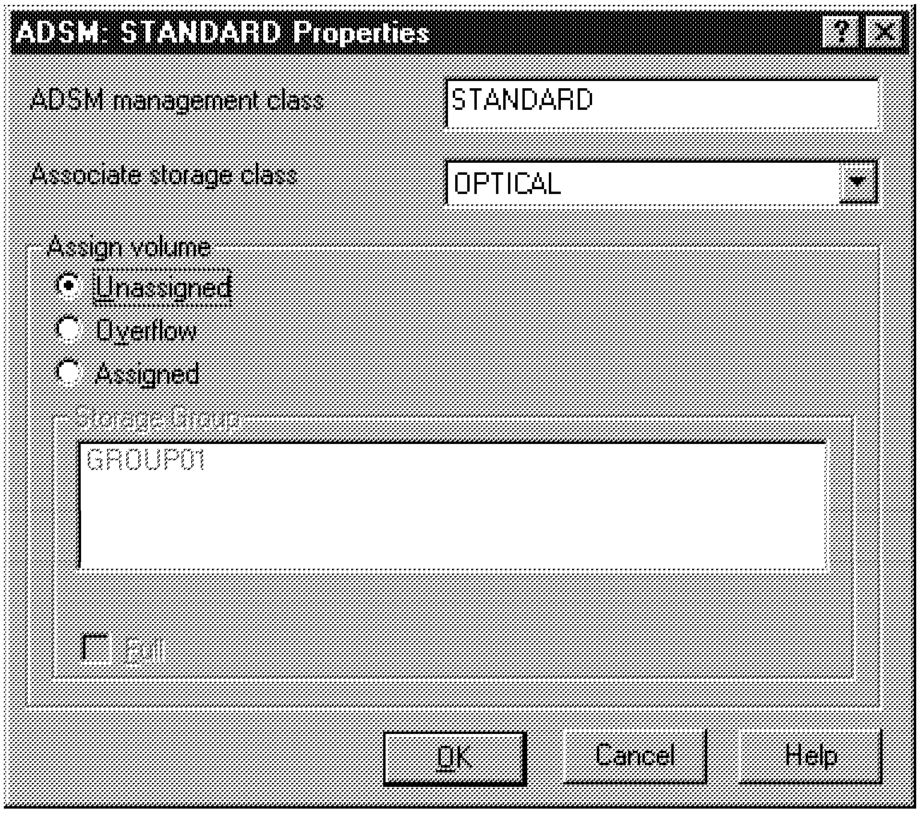
☑ Enable

Figure 52. FRNNADDM Properties Window

11. Create a new storage class, OPTICAL. Assign it the FRNNADDM device manager (the device manager for ADSM).

12. Create a new management class, MGTCLAS1, and associate the storage class OPTICAL with the default retention period FOREVER. This value is used for tests only.

13. Create a new volume of type ADSM. In the ADSM management class field, write STANDARD. Remember that the name of a VisualInfo ADSM volume must match the name of an ADSM management class. Associate the OPTICAL storage class with this volume (see Figure 53 on page 75).



ADSM: STANDARD Properties

ADSM management class: STANDARD

Associate storage class: OPTICAL

Assign volume

● Unassigned

○ Overflow

○ Assigned

Storage Group: GROUP01

□ Full

Figure 53. ADSM:STANDARD Properties Window

The SMS server links to ADSM when creating the ADSM volume. This is the first stage at which the link is tested, so you may get problems here. Refer to B.1.2, "Log Files" on page 192 if you encounter problems.

14. Create a new storage group with name GROUP02. Associate the volume ADSM:STANDARD.

15. Create a new collection with name CBR.CLLCT002, management class OPTLIB, and storage group GROUP02.

16. Edit the staging area and change the maximum size to 10 MB, the start purge to 10%, and the end purge to 1%. These values are for test purposes only. With these values you can work with a few objects and, when you run the destager, all of the objects are moved to optical.

17. Create a new index class with the same properties as the NOINDEX class. Select the NOINDEX class, click with the left mouse button, and select **Create from.** Name the class TESTOPTICAL, with abbreviation TESTO, and select the **CBR.CLLCT002** collection in the Class Servers tab. In the Class Processing tab, deselect Workflow processing.

18. Exit the application.

19. Start the object server. It will link to ADSM to validate its defined volumes.

20. Start the client application and index some documents into the newly defined class TESTOPTICAL. The objects are stored in the staging area. Run the destager from the object server window to write the objects to ADSM.

5.8 Configuring Example 1

In this section we describe how to configure a system to implement example 1. See 4.1, “Example 1: Windows NT” on page 53 for a description of the business requirements of this example. Our main purpose is to show you how to configure System Managed Storage, including the ADSM configuration. We do not cover how to define and create users, access control lists, or index classes.

5.8.1 Creating ADSM Objects

These are the steps to follow to create the ADSM objects:

1. Policy domain and policy set

   Use the STANDARD policy domain and the STANDARD policy set that are installed by default during the ADSM installation. Remember that you can have only one policy set active at a time.

2. Management class

   Define two new management classes to store the objects for the example:

   ```
   adsm> define mgmtclass standard standard example1 desc=′VI volume 1′
   adsm> define mgmtclass standard standard example2 desc=′VI volume 2′
   ```

   These classes have the same name as the VisualInfo ADSM volumes we will define later. The name of the classes are EXAMPLE1 and EXAMPLE2. STANDARD is the name of the policy domain and the name of the policy set to which the management class belongs. You have now these management classes:

   ```
   adsm> q mgmtclass

   Policy        Policy        Mgmt          Default       Description
   Domain        Set Name      Class         Mgmt
   Name                        Name          Class ?
   ---------     ---------     ---------     ---------     ----------------------
   STANDARD      ACTIVE        STANDARD      Yes           Installed default
                                                            management class.
   STANDARD      STANDARD      EXAMPLE1      No            VI volume 1
   STANDARD      STANDARD      EXAMPLE2      No            VI volume 2
   STANDARD      STANDARD      STANDARD      Yes           Installed default
                                                            management class.
   ```

3. Storage pools

   Define two new storage pools, SAMPLE1 and SAMPLE2, to be used with the optical library:

   ```
   adsm> define stgpool sample1 c64opt maxscratch=10 desc=′Stg pool for
   VI R/W′
   adsm> define stgpool sample2 c64opt maxscratch=10 desc=′Stg pool for
   VI WORM′
   ```

   These new storage pools are assigned to device class c64opt. By default they are primary storage pools, with READWRITE access, and with no limit on maximum file size. The maximum number of scratch volumes that can be requested for these storage pools is 10 (you can update this number if you need to). You now have these storage pools:

```
adsm> q stgpool

Storage       Device       Estimated     Pct     Pct   High   Low   Next
Pool Name     Class Name    Capacity    Util    Migr   Mig    Mig   Storage
                                (MB)                    Pct   Pct   Pool
-----------   ----------   ----------   -----   -----   ----   ---   -----------
ARCHIVEPOOL   DISK                0.0     0.0     0.0     90    70
BACKUPPOOL    DISK                4.0    87.5    87.4     90    70   C640PTPOOL
C640PTPOOL    C640PT         24,863.5     0.1    10.0     90    70
SAMPLE1       C640PT              0.0     0.0     0.0     90    70
SAMPLE2       C640PT              0.0     0.0     0.0     90    70
SPACEMGPOOL   DISK                0.0     0.0     0.0     90    70
```

You define two storage pools because you will use WORM disks in one of them and R/W disks in the other.

4. Copy group

   Define two new backup copy groups for each management class:

   ```
   adsm> define copygroup standard standard example1 type=backup
   destination=sample1 verexists=1 verdeleted=0
   adsm> define copygroup standard standard example2 type=backup
   destination=sample2 verexists=1 verdeleted=0
   ```

   Where:

   - *destination* is the name of the primary storage pool where backup data is to be stored.
   - *verexists* is the maximum number of backup versions to retain for files that are currently on the client file system. For VisualInfo, this value must be 1.
   - *verdeleted* is the maximum number of backup versions to retain for files that are deleted from the client file system. For VisualInfo this number must be 0.

   You now have these copy groups:

```
adsm> query copygroup

Policy    Policy    Mgmt      Copy      Versions Versions   Retain Retain
Domain    Set Name  Class     Group         Data     Data    Extra   Only
Name                Name      Name        Exists  Deleted Versions Version
--------- --------- --------- --------- -------- -------- -------- -------
STANDARD  ACTIVE    STANDARD  STANDARD         2        1       30      60
STANDARD  STANDARD  EXAMPLE1  STANDARD         1        0       30      60
STANDARD  STANDARD  EXAMPLE2  STANDARD         1        0       30      60
STANDARD  STANDARD  STANDARD  STANDARD         2        1       30      60
```

   The Retain Extra Versions and Retain Only Version parameters are not used when you set Versions Data Exists to 1.

5. Now you must activate the policy set:

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

   When a policy set is activated, the contents of the policy set are copied to a policy set that has the reserved name ACTIVE. Once activated, there is no real relationship between the policy set that has been activated (copied to ACTIVE) and the contents of the ACTIVE policy set. The original policy set can still be modified, but the copied definitions in the ACTIVE policy set can be modified only by activating another policy set.

Because of this copying process, you do not assume that the definitions in the policy set that was last activated are the same as those in the ACTIVE policy set; the original policy set may have been modified since activation. These definitions have been activated:

```
adsm> q mgmtclass

Policy        Policy        Mgmt          Default       Description

Domain        Set Name      Class         Mgmt
Name                        Name          Class ?
---------     ---------     ---------     ---------     ------------------------
STANDARD      ACTIVE        EXAMPLE1      No            VI volume 1
STANDARD      ACTIVE        EXAMPLE2      No            VI volume 2
STANDARD      ACTIVE        STANDARD      Yes           Installed default
                                                         management class.
STANDARD      STANDARD      EXAMPLE1      No            VI volume 1
STANDARD      STANDARD      EXAMPLE2      No            VI volume 2
STANDARD      STANDARD      STANDARD      Yes           Installed default
                                                         management class.
```

6. Add these lines in the c:\win32app\ibm\adsm\api\dll\dsm.opt file:

   ```
   INCLUDE ?:EXAMPLE1* EXAMPLE1
   INCLUDE ?:EXAMPLE2* EXAMPLE2
   ```

   You have to add a new line for each non-default ADSM management class that you use in your system.

5.8.2 Creating VisualInfo SMS Objects

In this section, we show you the objects needed for implementing example 1 and the order in which to create them. You must have a clear idea of what you are doing in SMS, because you can end up with inconsistencies that prevent you from deleting some objects.

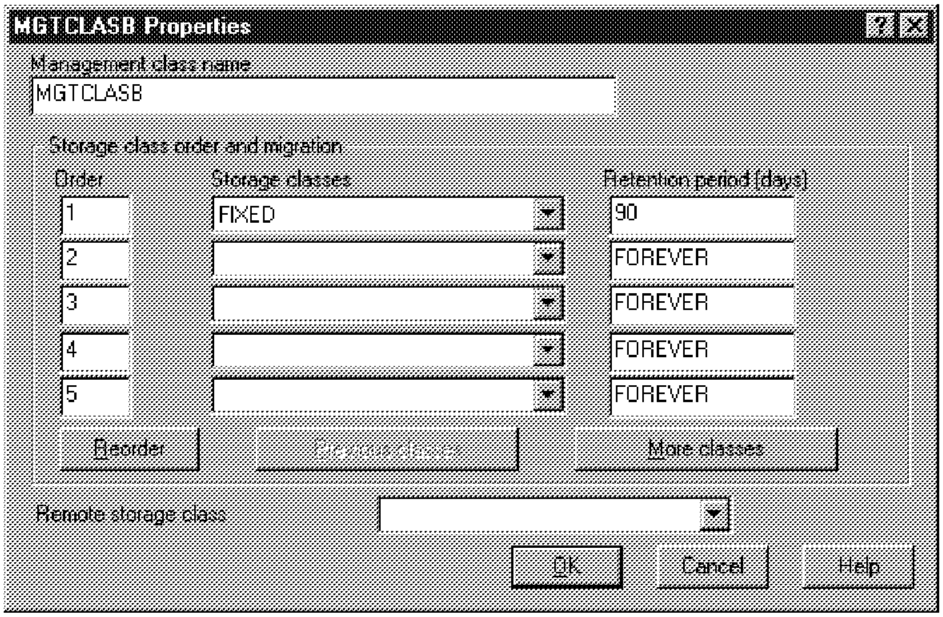
You need three volumes, one for DASD and two for the optical library. You also need three management classes. Then you define the storage groups and the collections.

Start the VisualInfo system administration program and follow these steps:

1. Create two new ADSM volumes, EXAMPLE1 and EXAMPLE2, and storage class OPTICAL. Leave them unassigned to any storage group. These are the volumes the system will use to store documents in the optical library.

2. Create three new management classes with the parameters shown in the following figures:

   Management class B is shown in Figure 54 on page 79,

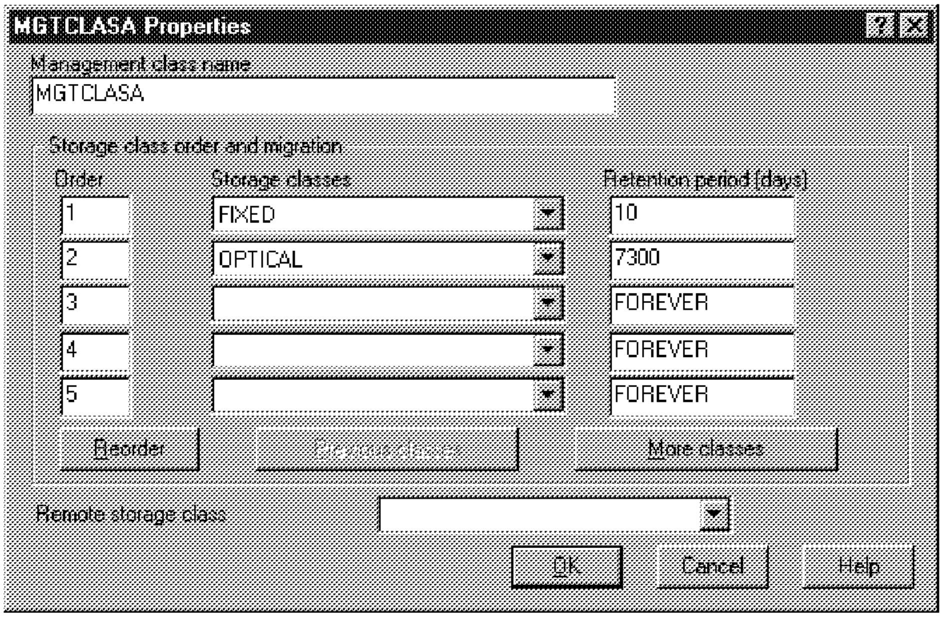


MGTCLASB

Order	Storage classes	Retention period (days)
1	FIXED	90
2		FOREVER
3		FOREVER
4		FOREVER
5		FOREVER

Figure 54. Properties of Management Class MGTCLASB

Management class A is shown in Figure 55.

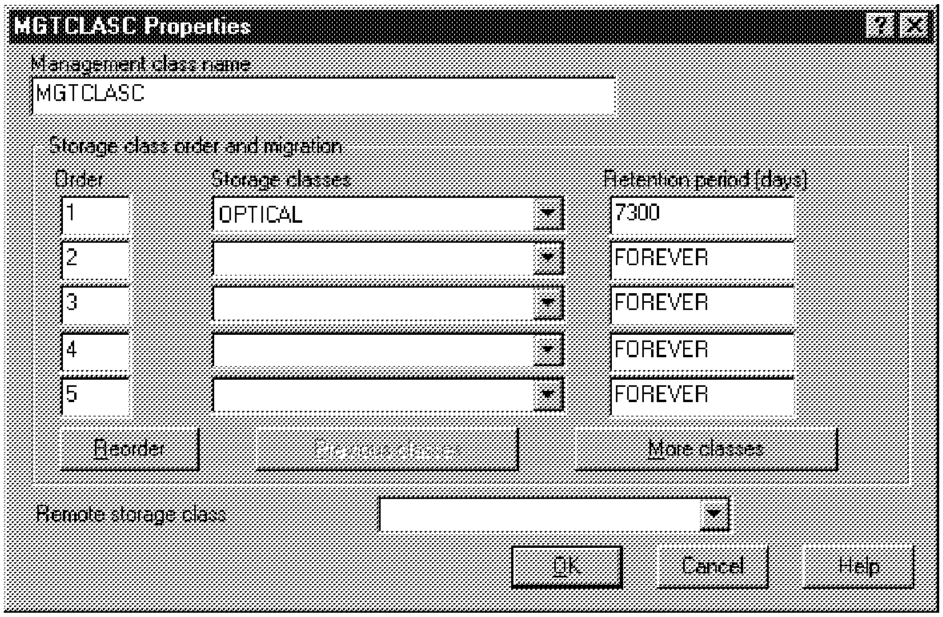


MGTCLASA

Order	Storage classes	Retention period (days)
1	FIXED	10
2	OPTICAL	7300
3		FOREVER
4		FOREVER
5		FOREVER

Figure 55. Properties of Management Class MGTCLASA

Management class C is shown in Figure 56 on page 80.



Management class name: MGTCLASC

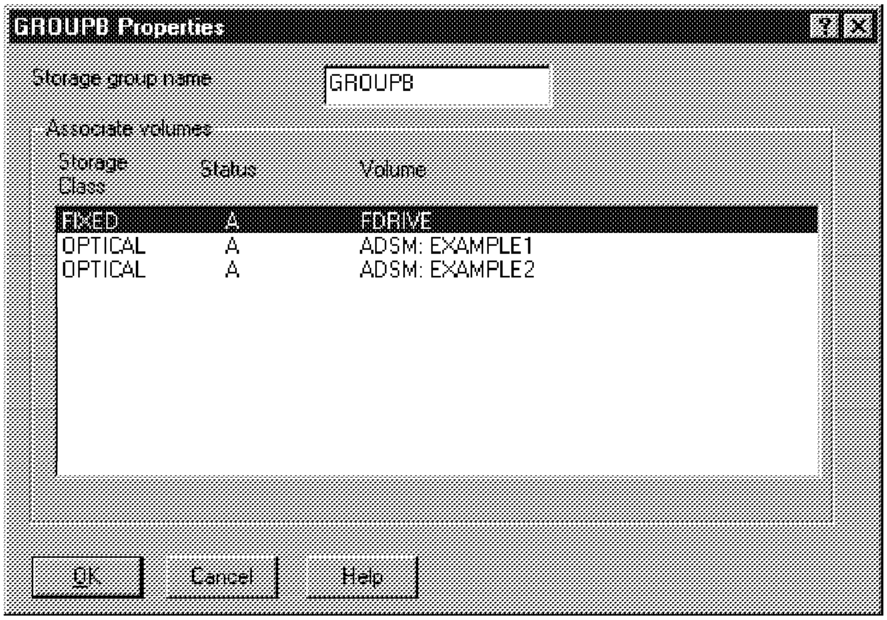
Remote storage class:

Order	Storage classes	Retention period (days)
1	OPTICAL	7300
2		FOREVER
3		FOREVER
4		FOREVER
5		FOREVER

Figure 56. Properties of Management Class MGTCLASC

Create three storage groups with the properties shown in the following figures:

Storage group B Figure 57.

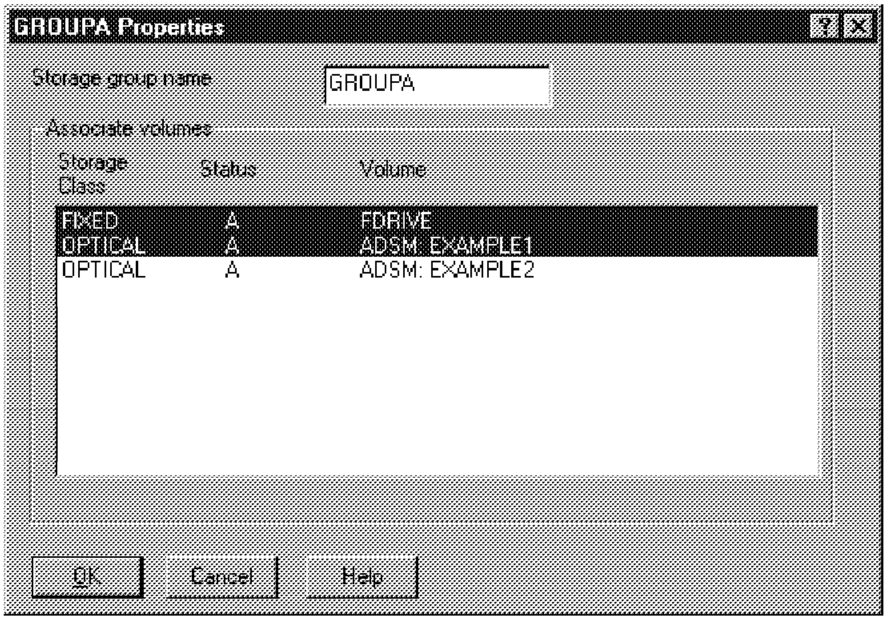


Storage group name: GROUPB

Storage Class	Status	Volume
FIXED	A	FDRIVE
OPTICAL	A	ADSM: EXAMPLE1
OPTICAL	A	ADSM: EXAMPLE2

Figure 57. Properties of Storage Group GROUPB

Storage group A is shown in Figure 58 on page 81.

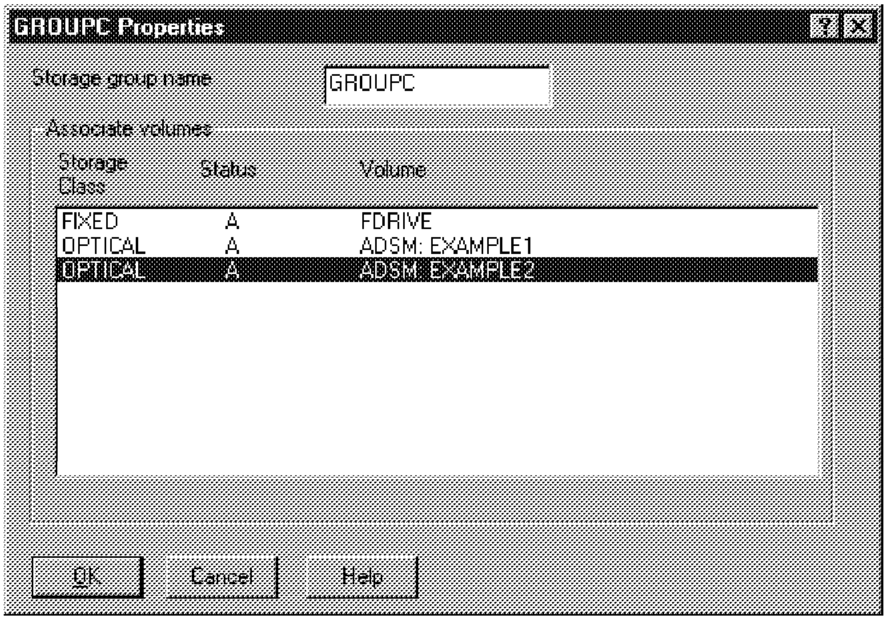


Storage group name: GROUPA

Storage Class	Status	Volume
FIXED	A	FDRIVE
OPTICAL	A	ADSM: EXAMPLE1
OPTICAL	A	ADSM: EXAMPLE2

Figure 58. Properties of Storage Group GROUPA

Storage group C is shown in Figure 59.



Storage group name: GROUPC

Storage Class	Status	Volume
FIXED	A	FDRIVE
OPTICAL	A	ADSM: EXAMPLE1
OPTICAL	A	ADSM: EXAMPLE2

Figure 59. Properties of Storage Group GROUPC

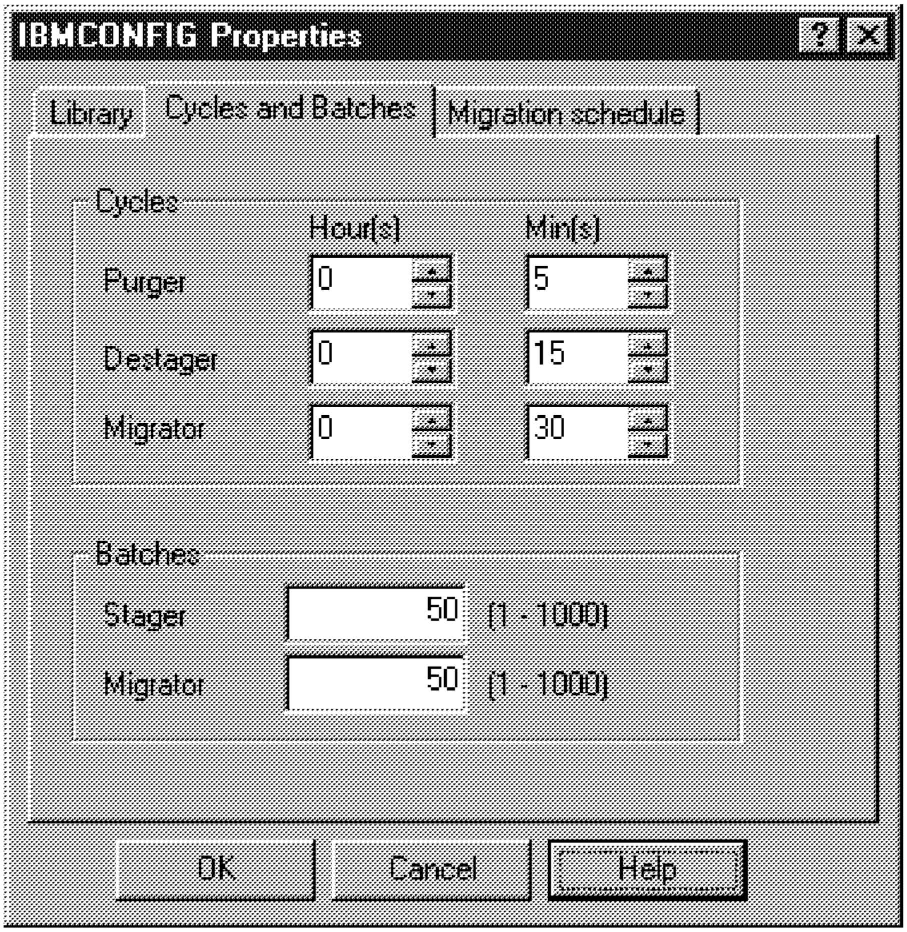
3. Create three new collections:

   - CBR.CLLCTB, with management class MGTCLASB and storage group GROUPB
   - CBR.CLLCTA, with management class MGTCLASA and storage group GROUPA
   - CBR.CLLCTC, with management class MGTCLASC and storage group GROUPC

4. Change the values for the staging area. Set the maximum size to 200 MB, the start purge rate to 80%, and the stop purge rate to 50%. These values ensure that the staging area will not be filled, and that no destaging is done during working hours.

5.8.3 Cycles

It is important to decide when the destager and the purger will run. The ideal situation is to have a large enough staging area that no destaging will occur during working hours, because the process can slow down the system. Figure 60 shows the values for example 1.



Cycles

	Hour(s)	Min(s)
Purger	0	5
Destager	0	15
Migrator	0	30

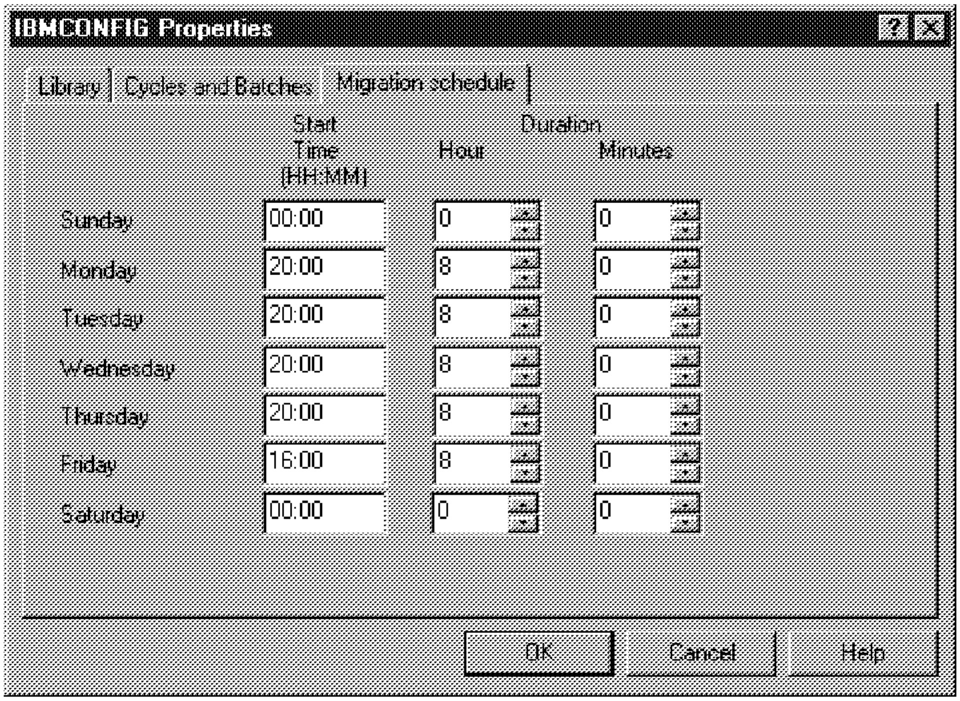
Batches

Stager	50	[1 - 1000]
Migrator	50	[1 - 1000]

Figure 60. Cycles and Batches

5.8.4 Migration Schedule

You must define the timeframe in which you want the migrator to work. You define the day of the week, the time to start, and the duration. In our case, we do not want the migrator to run on Saturday and Sunday, and we want to leave it running for no more than 8 hours at night. See Figure 61 on page 83.



	Start Time (HH:MM)	Duration Hour	Duration Minutes
Sunday	00:00	0	0
Monday	20:00	8	0
Tuesday	20:00	8	0
Wednesday	20:00	8	0
Thursday	20:00	8	0
Friday	16:00	8	0
Saturday	00:00	0	0

Figure 61. Migration Schedule

5.8.5 Index Classes and Key Fields

Now you must create the key fields and the index classes for your system, and assign the correct collection to each index class. You can have as many index classes as you need, but you have to assign one of these collections to each index class.

Chapter 6. Setting Up Example 2 on AIX

In this chapter, we cover the installation, setup and configuration of a VisualInfo subsystem on an AIX platform. We also cover the installation and the configuration of prerequisite subsystems.

We do not cover the installation and configuration of an AIX operating system and of the subsystems provided with AIX (TCP/IP, X11), nor the installation and configuration of the Java Development Toolkit (JDK).

6.1 Configuration for Example 2

Figure 62 shows the setup for installing and configuring the necessary subsystems and their components to accommodate the SMS requirements detailed in Example 2. This example is described in 4.2, "Example 2: AIX" on page 57.

AIX platform ROLM3 is the VisualInfo server, including an ADSM client.

AIX platform ROLM7 is the ADSM server, connected to the optical library.

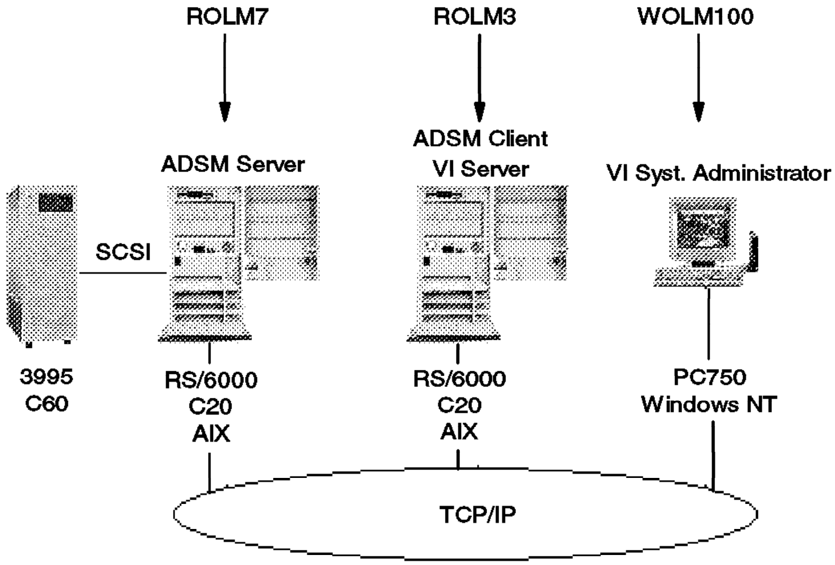


Figure 62. Setup for Example 2

6.2 Example 2 Hardware and Software

In this section we list the hardware and software selected for setting up Example 2 (see Figure 62). All VisualInfo hardware and software requirements are fulfilled.

6.2.1 Hardware

1. **ADSM server**
    - RS/6000: model C20
    - RAM: 132MB
    - DASD: 3 x 2GB
    - Paging space: 192MB
2. **VisualInfo server**
    - RS/6000 model C20
    - RAM: 132MB
    - DASD: 3 x 2GB
    - Paging space: 160MB
    - Display: VGA
3. **System Administrator**
    - PC 350 166MHz
    - RAM: 48MB
    - DASD: 1 x 1GB
4. **Optical Library**
    - IBM 3995 model C60
    - Optical drives: 2
    - Maximum slot: 20

6.2.2 Software

1. **ADSM server**
    - AIX version 4.2.1
    - ADSM server version 3.1.0.1 (U452223)
    - ADSM client version 3.1.20
2. **VisualInfo server**
    - AIX version 4.2.1
    - ADSM client version 3.1.20
    - C Set ++ version 3.1.4 (U451810)
    - Java Development Tool Kit version 1.1.2
    - VisualInfo server version 2.3.1
    - DB2/6000 version 2.1.1 (U441267 and U444032)
3. **System Administrator**
    - Windows NT Server: version 4.00.1381
    - VisualInfo system administration: version 2.3.1

6.3 Installation and Configuration of Prerequisite Software

In this section we explain how to install and configure the prerequisite software for VisualInfo.

1. DB2/6000
2. C Set ++
3. ADSM

In our environment (X-server), we use the *.profile* file to define the profile of users. If you work with the Common Desktop Environment (CDE), use the *.dtprofile* file instead to define the user profiles.

6.3.1 Installation of DB2/6000

DB2/6000 is required in order to store objects in VisualInfo. Follow these steps to install DB2/6000:

Note: The first two steps are only necessary if you are using the CD-ROM drive for the first time.

1. Log in as the root user and do not run any other process until you are finished with your installation.
2. Create a directory called cdrom under root, then create a cdrom file system using the SMIT interface.

   ```
   # smit crfs
   ```

   Select **add CD-ROM filesystem**, enter the CD-ROM device name (for example cd0) from the pop-up menu by pressing the F4 key, select **mount point over root /** and press enter.
3. Insert the CD-ROM for DB2 and mount it:

   ```
   # mount /dev/cd0 /cdrom
   ```

4. Enter this command to begin the installation:

   ```
    # smit install
   ```

5. From the displayed menu, select **Install and Update from latest available software.**
6. Select the input device **/dev/cd0**.
7. For the **Software to install** field, expand the list by pressing the F4 key.
8. Select the following components:

```
-DB2 client Application Enabler
-DB2 Command Line Processor
-DB2 Communications Support Base with TCPIP
-DB2 Database Director
-DB2 Runtime executables
-DB2 Product Library inf or ps in English (OPTIONAL)
-DB2 SDK Utilities and Samples
-DB2 Utilities and Samples
-DB2 Visual Explain
-DDCS DRDA Application Requestor
-Install all DB2 client components
-Install all SDK Components
-Install Single user Components.
```

9. Press Enter.
10. Set the option for installing prerequired software to YES.
11. Press Enter twice to begin the installation.

After this installation is complete, you have to apply PTFs U441267 and U444032 for DB2/6000 2.1.1. Then your DB2/6000 2.1.1 installation for VisualInfo 2.3.1 is complete.

6.3.2 Installation of C Set ++

Using the same CD-ROM file system, unmount the DB2 CD-ROM device filesystem. Insert the C Set ++ CD-ROM and mount it. Follow these steps to install C Set ++:

1. Enter this command:

   ```
   # smit install
   ```

2. Select **Install and Update from Latest available software**.
3. Specify the input device.
4. Select all components to install.
5. Set the option of automatic install pre-required components to YES.
6. Press Enter twice to begin the installation.
7. After a successful installation of C Set ++, you need to apply PTF U451810 for C Set ++ 3.1.4 to be in compliance with the level and prerequisites of VisualInfo.
8. To ensure that the C Set ++ compiler is licensed, run the sample program from the /usr/lpp/xlC/tutorial/source directory as:

   ```
   # xlC 4-1.C
   ```

   where 4-1.C is the C Set ++ program. If the compilation is successful, you get a command prompt without any messages. If you get a `License Failure` or `License Expiration` message, you have to register the product. Contact your administrator.

9. We recommend installing VisualInfo only after you have the C Set ++ license.

6.3.3 Installation and Configuration of ADSM

In this section we explain how to install and configure the ADSM subsystem on both the AIX (ROLM3 and ROLM7) platforms.

- ADSM server and ADSM client on ROLM7
- ADSM client on ROLM3

Note: We have detailed the ADSM configuration steps in the following sections, but it is strongly recommended that you have some experience with ADSM before executing them.

The ADSM server manages an IBM 3995 Optical Library Dataserver, so we include the different steps to configure the library and define it on the AIX system.

6.3.3.1 Installation of the Optical Library

In our configuration we use an IBM 3995 Optical Library Dataserver model C60, which provides high-capacity online storage. When ordering an IBM 3995 model, be sure to specify in which environment it is used (for example, attached to a LAN, or an RS/6000, or an AS/400). Each model is preloaded with specific microcode related to the attachment used.

Applying a Label to the Optical Disk Platters: An optical disk platter has two sides, A and B. ADSM sees the platter as a *volume*, so it is not possible to separate each side. Before using any optical disk platter, you must identify it using the label provided with it. In our configuration, we use four rewritable optical disk platters identified as:

- VI001A
- VI002A
- VI003A
- VI004A

In the configuration steps we explain how to insert platters into the optical library, format them and register them in ADSM.

Defining the SCSI Address: Our optical library contains:

- One autochanger
- Two optical drives

The autochanger SCSI address is usually set to 6. Drive 1 is usually set to SCSI address 1, and drive 2 is usually set to SCSI address 2. We kept these default addresses.

However, different SCSI addresses may be required by the application controlling the library or by the RS/6000 configuration. To change addresses, consult *IBM 3995 Optical Library Dataserver: Maintenance Information for Model Cn0*, Part Number 05h5422.

Connecting the Optical Library to the RS/6000: If the optical library and the RS/6000 are powered off, connect both of them with the SCSI cable provided.

Important

A single-ended SCSI terminator must be installed in the SCSI connector below the redriver board. Otherwise, even if the related drives are available for AIX, the storage of files in the platters fails randomly.

Then power on the optical library and the RS/6000. Both drives installed in the optical library should be available for the AIX system. To verify, log on to the AIX system with the root user ID and execute the following command:

```
# lsdev -C -c rwoptical
```

The output shows the optical drives found and their states:

```
omd0 Available 00-01-01-1,0 SCSI Read/Write Optical Drive
omd1 Available 00-01-01-2,0 SCSI Read/Write Optical Drive
```

The subaddresses of the drive devices (1 and 2) on SCSI bus 00-01-01 are set by default in the optical library. Keep them in mind as you will be asked for them when you configure the ADSM resources.

6.3.3.2 Installation of ADSM on the AIX Platform

In this section, we explain how to install and configure the ADSM server and ADSM client on the AIX platform.

Installation of ADSM Server and ADSM Client on ROLM7: Installing the ADSM server and ADSM client on AIX (ROLM7) is very simple. Use SMIT to execute this operation.

Important

ADSM client module adsm.client.aix4.2 requires Level 4.2.1 of AIX module bos.rte so we installed the latest version of AIX (4.2.1) before we installed ADSM modules. Otherwise, installation of the ADSM client module would fail.

Insert the ADSM for AIX CD-ROM into the CD-ROM drive and log on with the root user ID. Then execute the following command:

```
# smit install
```

When the System Management Interface Tool: root@rolm7 window appears, select successively the following options:

```
Install and Update Software
    --> Install and Update from Latest Available Software
```

When the Install and Update from Latest Available Software: root@rolm7 window appears, select the following option:

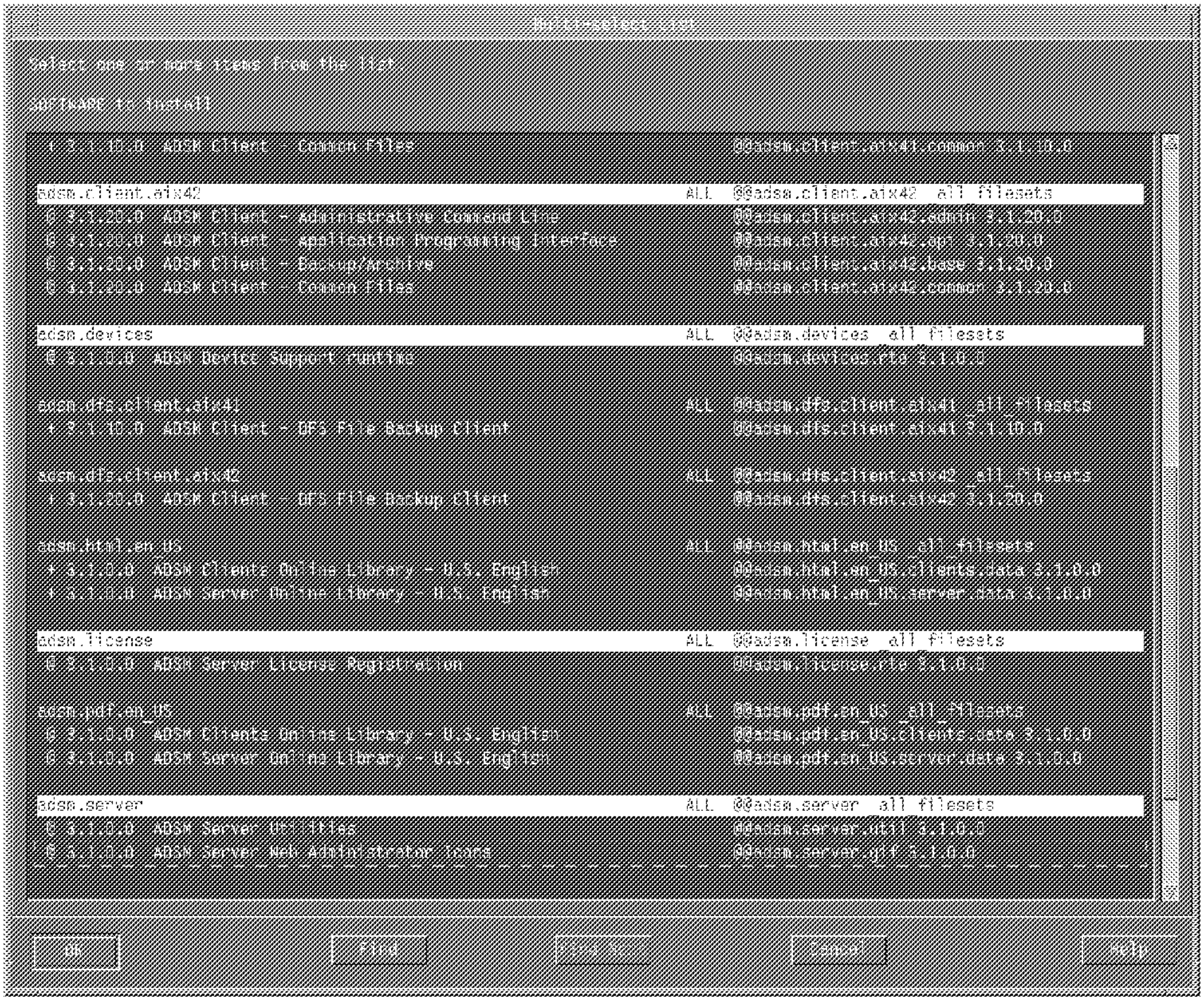
```
INPUT device  --> /dev/cd0
```

Then click on **OK**.

When the Install and Update from Latest Available Software: root@rolm7 window appears, click on the Software to Install **list** and select the following components:

```
SOFTWARE to install --> adsm.client.aix42
                    --> adsm.devices
                    --> adsm.licence
                    --> adsm.server
```

Figure 63 shows the window displayed once the choices have been made.



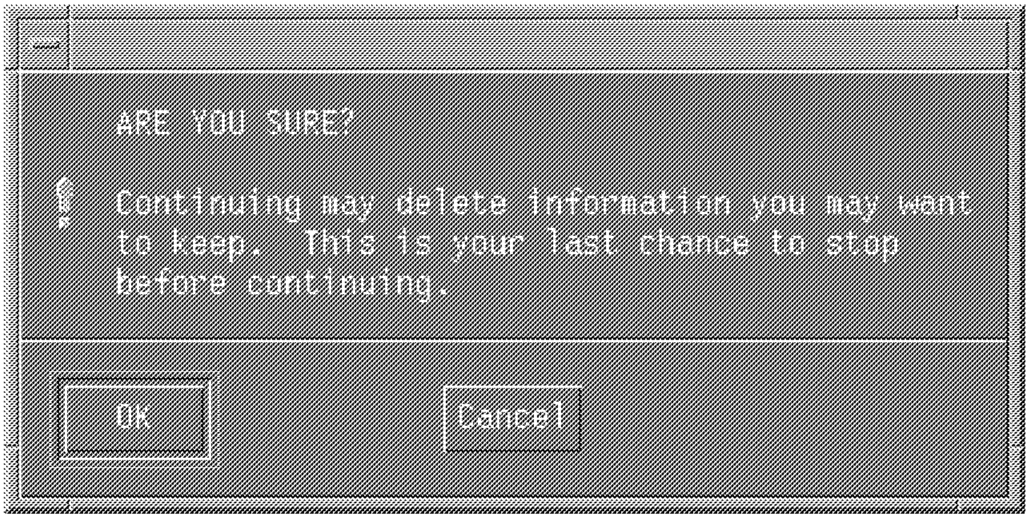
Select one or more items from the list.

SOFTWARE to install

```
  + 3.1.10.0  ADSM Client - Common Files                         @@adsm.client.aix41.common 3.1.10.0

adsm.client.aix42                                           ALL  @@adsm.client.aix42  all filesets
  @ 3.1.20.0  ADSM Client - Administrative Command Line          @@adsm.client.aix42.admin 3.1.20.0
  @ 3.1.20.0  ADSM Client - application Programming Interface    @@adsm.client.aix42.api 3.1.20.0
  @ 3.1.20.0  ADSM Client - Backup/Archive                       @@adsm.client.aix42.base 3.1.20.0
  @ 3.1.20.0  ADSM Client - Common Files                         @@adsm.client.aix42.common 3.1.20.0

adsm.devices                                                ALL  @@adsm.devices  all filesets
  @ 3.1.0.0  ADSM Device Support runtime                         @@adsm.devices.rte 3.1.0.0

adsm.dfs.client.aix41                                       ALL  @@adsm.dfs.client.aix41  all filesets
  + 3.1.10.0  ADSM Client - DFS File Backup Client               @@adsm.dfs.client.aix41 3.1.10.0

adsm.dfs.client.aix42                                       ALL  @@adsm.dfs.client.aix42  all filesets
  + 3.1.20.0  ADSM Client - DFS File Backup Client               @@adsm.dfs.client.aix42 3.1.20.0

adsm.html.en_US                                             ALL  @@adsm.html.en_US  all filesets
  + 3.1.0.0  ADSM Clients Online Library - U.S. English          @@adsm.html.en_US.clients.data 3.1.0.0
  + 3.1.0.0  ADSM Server Online Library - U.S. English           @@adsm.html.en_US.server.data 3.1.0.0

adsm.license                                                ALL  @@adsm.license  all filesets
  @ 3.1.0.0  ADSM Server License Registration                    @@adsm.license.rte 3.1.0.0

adsm.pdf.en_US                                              ALL  @@adsm.pdf.en_US  all filesets
  @ 3.1.0.0  ADSM Clients Online Library - U.S. English          @@adsm.pdf.en_US.clients.data 3.1.0.0
  @ 3.1.0.0  ADSM Server Online Library - U.S. English           @@adsm.pdf.en_US.server.data 3.1.0.0

adsm.server                                                 ALL  @@adsm.server  all filesets
  @ 3.1.0.0  ADSM Server Utilities                               @@adsm.server.util 3.1.0.0
  @ 3.1.0.0  ADSM Server Web Administrator Icons                 @@adsm.server.gif 3.1.0.0
```

Figure 63. Installing ADSM: Components Selected for Installation

Click on **OK** to validate them. The next window displayed, Figure 64 on page 92, asks you to confirm the start of this installation.



ARE YOU SURE?

Continuing may delete information you may want to keep. This is your last chance to stop before continuing.

Figure 64. Accepting the Start of ADSM Installation

Click on **OK** to proceed with the installation. Wait until the installation is completed successfully, then exit the SMIT interface.

The installation procedure modifies the /etc/inittab file by inserting a line that will have the ADSM server start ADSM automatically each time the server is rebooted. For convenience during our test, we removed that line. In order not to edit this critical file, we used the following command:

```
# dsm_rmv_itab autostart
```

At the end of the installation, we execute the following command to register the start of the ADSM server with the ROLM7.

```
# dsm_add_itab autostart
```

When the ADSM server is started from the /etc/inittab file, it uses the quiet option. No console window is displayed. The only way to stop it is to run an administration client interface and to execute the following command:

```
halt
```

Install the ADSM Client on AIX: To install the ADSM client on AIX (ROLM3), follow the procedure described in “Installation of ADSM Server and ADSM Client on ROLM7” on page 90. Only the requested modules, adsm.client.aix42, should be selected for installation.

Update the ADSM Server on ROLM7: Service Level U452223 for ADSM version 3.1 for AIX is available; we applied it. It contains corrections to the server run-time file set and the server NLS message repository and online help.

Update Tip

This update allows a workstation running a Web browser to access the ADSM server and to administer it. It is useful because the administrative client GUI is no longer provided with ADSM version 3.1 on the AIX platform.

Basic Configuration of the ADSM Server and ADSM Client on ROLM7: On the ADSM server ROLM7, basic configurations should be defined for both the ADSM server and client:

1. Basic configuration for ADSM server

   Basic ADSM server parameters are stored in the ADSM server file, /usr/lpp/adsmserv/bin/dsmserv.opt. The easiest way to configure this file is to use the ADSM server configuration file sample provided in the /usr/lpp/adsmserv/bin/dsmserv.opt file and modify each entry according to your configuration. We modified the following parameters for our configuration:

```
COMMmethod SHARE
COMMmethod TCPIP
SHMPort 1510
TCPPort 1500
TCPWindowsize 16
MAXSessions 25
VOLUMEHistory  /usr/lpp/adsmserv/bin/volhist
DEVCONFig /usr/lpp/adsmserv/bin/devconfig
```

   The ADSM server will use the Shared Memory protocol to communicate with the ADSM client installed in the same machine (ROLM7).

   It will use TCP/IP to communicate with ROLM3 through the local network.

   Once the basic configuration is complete and the related files are stored, start the ADSM server by executing the following commands from the /usr/lpp/adsmserv/bin directory:

```
# dsmserv
```

   The ADSM server starts and indicates the state of its resources:

```
ANR0900I Processing options file dsmserv.opt.
ANR0990I ADSM server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 8 megabytes.
ANR0201I Database assigned capacity is 12 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR2100I Activity log process has started.
ANR1305I Disk volume /usr/lpp/adsmserv/bin/backup.dsm varied online.
ANR1305I Disk volume /usr/lpp/adsmserv/bin/archive.dsm varied online
ANR1305I Disk volume /usr/lpp/adsmserv/bin/spcmgmt.dsm varied online.
ANR0984I Process 1 for EXPIRATION started in the BACKGROUND at 12:46:11.
ANR0811I Inventory client file expiration started as process 1.
ANR0812I Inventory file expiration process 1 completed: examined 0 object
         0 objects, deleting, 0 backup objects, 0 archive objects,
         and 0 DB backup volumes. 0 errors were encountered.
ANR0985I Process 1 for EXPIRATION running in the BACKGROUND completed
          with completion state SUCCESS at 12:46:11.
ANR2803I License manager started.
ANR82001 TCP/IP driver ready for connection with clients on port 1500.
ANR8285I Shared Memory driver ready for connection with clients on
         port 1510
ANR2835I server is licensed for 1 clients
ANR2854I server is licensed for device support module 2
ANR0993I ADSM server initialization complete.
```

   To stop the ADSM server, execute the following command from the ADSM console or from an administration client interface:

```
adsm> halt
```

2. Basic configuration for ADSM client

   Basic ADSM client parameters are stored in ADSM server files /usr/lpp/adsm/bin/dsm.opt and /usr/lpp/adsm/bin/dsm.sys. These files are created during the installation process and include some parameters that you specify according to your configuration.

```
//ADSM_LOCAL_SERVER_STANZA
Servername         ADSM
SHMPort            1510
//END_ADSM_LOCAL_SERVER_STANZA
```

Figure 65. dsm.sys File for Local ADSM Client

   Figure 65 and Figure 66 show the parameters we used.

```
virtualnode        client
Servername         ADSM
```

Figure 66. dsm.opt File for Local ADSM Client

   For performance reasons we chose the Shared Memory protocol for the ADSM server and the ADSM client installed on ROLM7.

Basic Configuration for ADSM Client on ROLM3: On ROLM3, a basic configuration should be defined for the ADSM client. Parameters for this configuration are stored in files /usr/lpp/adsm/bin/dsm.opt and /usr/lpp/adsm/bin/dsm.sys. These files are created during the installation process and include some parameters that you have to specify according to your configuration.

```
Servername        ADSM
Nodename          rolm3
CommMethod        TCPIP
PasswordAccess    generate
TCPPort           1500
TCPServeraddress  9.37.195.225
```

Figure 67. dsm.sys File for Network ADSM Client

Figure 67 and Figure 68 show the parameters we used.

```
Servername         ADSM
```

Figure 68. dsm.opt File for Network ADSM Client

Here we had no choice other than to use TCP/IP.

Communication Tests: To ensure that the communication configurations have been set correctly, start one of the following AIX commands (which require communication between the ADSM server on ROLM7 and the ADSM client on ROLM3) and verify that they complete successfully:

1. Start the administrative client command interface.

```
# dsmadmc
```

When required, enter the ADSM administrator user ID (admin by default) and password (admin by default). You should receive the ADSM prompt (adsm>).

2. Start the GUI backup/restore application.

```
# dsm
```

When required, enter the ADSM client user ID (rolm3) and password (rolm3). You should access the GUI backup/restore application.

The node ROLM3 has been previously added in the ADSM server domain.

6.3.3.3 Advanced Configuration of the ADSM Server

In this section we describe the main operations for configuring the ADSM Server on ROLM7 for implementing Example 2. The ADSM resources we need for the environment described in 4.2.4, "ADSM, VisualInfo Fileroom and SMS Component Definitions" on page 59 are be defined in the following order:

1. Policy domain
2. Policy set
3. Management class
4. Copy group
5. Library
6. Drive
7. Device classes
8. Storage pool
9. Volumes

The library, drive and device classes are related to the hardware used, so they are defined only once. The other resources are configuration-dependent, so they are defined for each new configuration. To create them, we used the ADSM administrator commands listed in the *ADSTAR Distributed Storage Manager for AIX V 3.1 Administrator's Reference*.

When the ADSM server is installed, some resources are created automatically; we used only the policy domain named STANDARD. All other resources needed for our configuration are listed in Table 4.

Table 4. ADSM Resources to Define

ADSM Resource	**Name**
Policy Domain	STANDARD
Policy Set	PSE2
Management Class	MCE2ST, MCE2LT
Library	LB0
Drive	DRV0, DRV1
Storage Pool	SPE2ST, SPE2LT
Volumes	VI001A, VI002A, VI003A, VI004A
Device Class	DCE2OPTI

Figure 69 on page 96 shows the links among these ADSM resources.

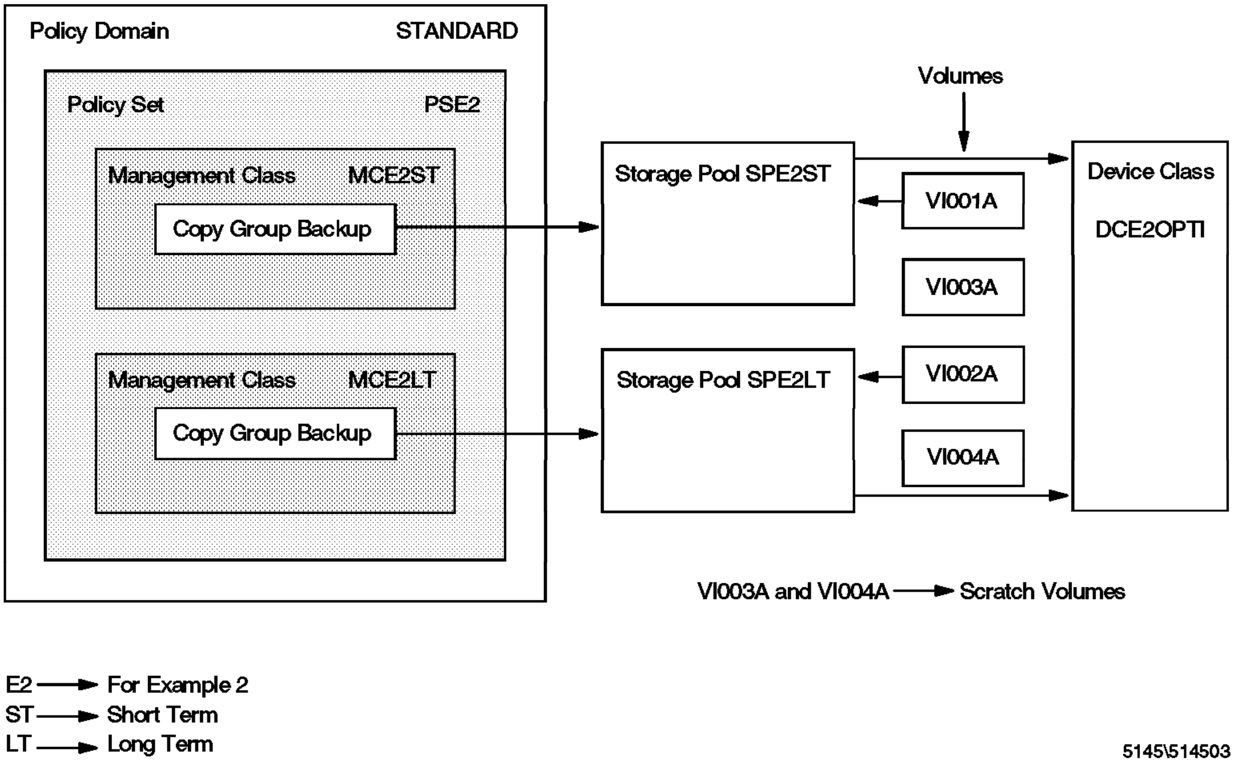


Figure 69. ADSM Resources for Example 2

Important

Before proceeding to configure any resources in ADSM, you must define on AIX (independently of the ADSM administration tasks) the library and its related drives.

Define the Library and its Drives in AIX: Once ADSM is installed, an ADSM device configuration entry is created in SMIT. Use it to configure the library and its optical drives.

1. Define the optical drives.

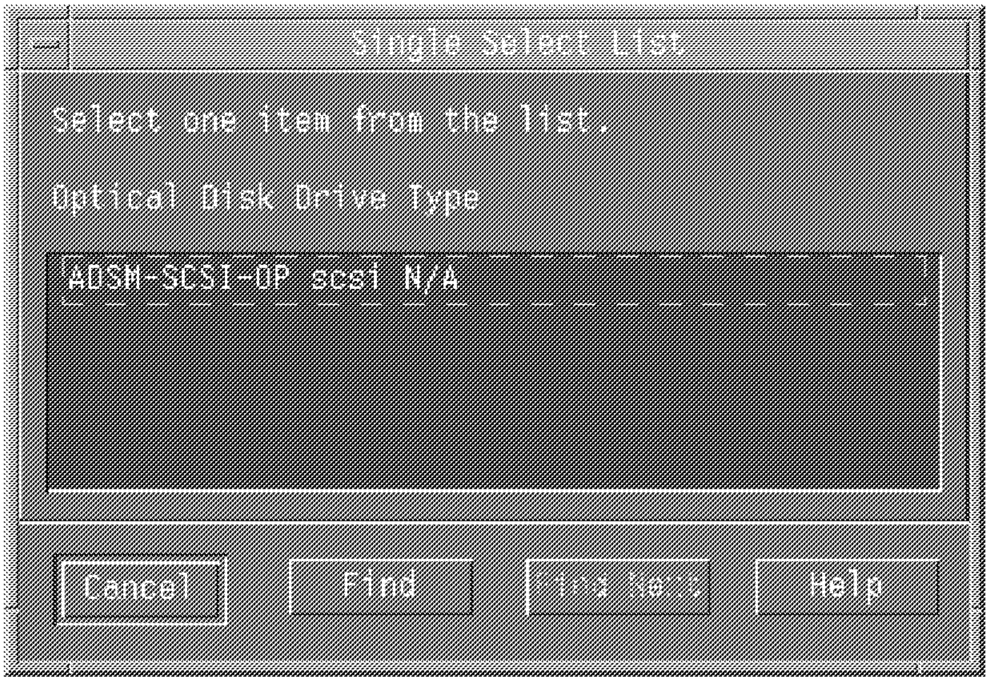
   Log on ROLM7 with the root user ID and execute the following command:

   ```
   # smit devices
   ```

   The System Management Interface Tool: root@rolm7 window appears; select:

   ```
   ADSM Devices
        --> Optical R/W Disk Drive
                --> Add a Optical Disk Drive
   ```

   The Single Select List window appears (see Figure 70 on page 97).



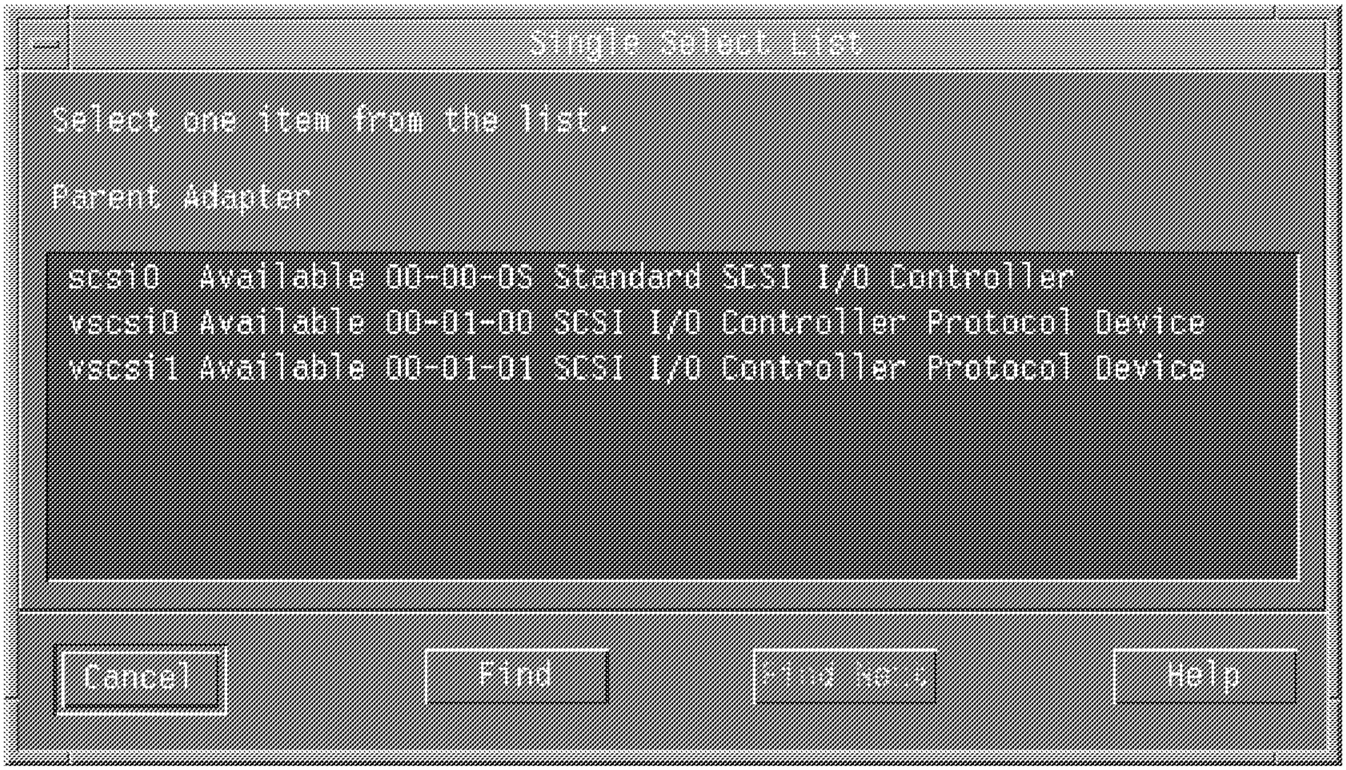
Select one item from the list.

Optical Disk Drive Type

```
ADSM-SCSI-OP scsi N/A
```

Figure 70. Optical Disk Drive Type Selection

Select the **ADSM-SCSI-OP scsi** entry. The Single Select List window appears (see Figure 71).



Select one item from the list.

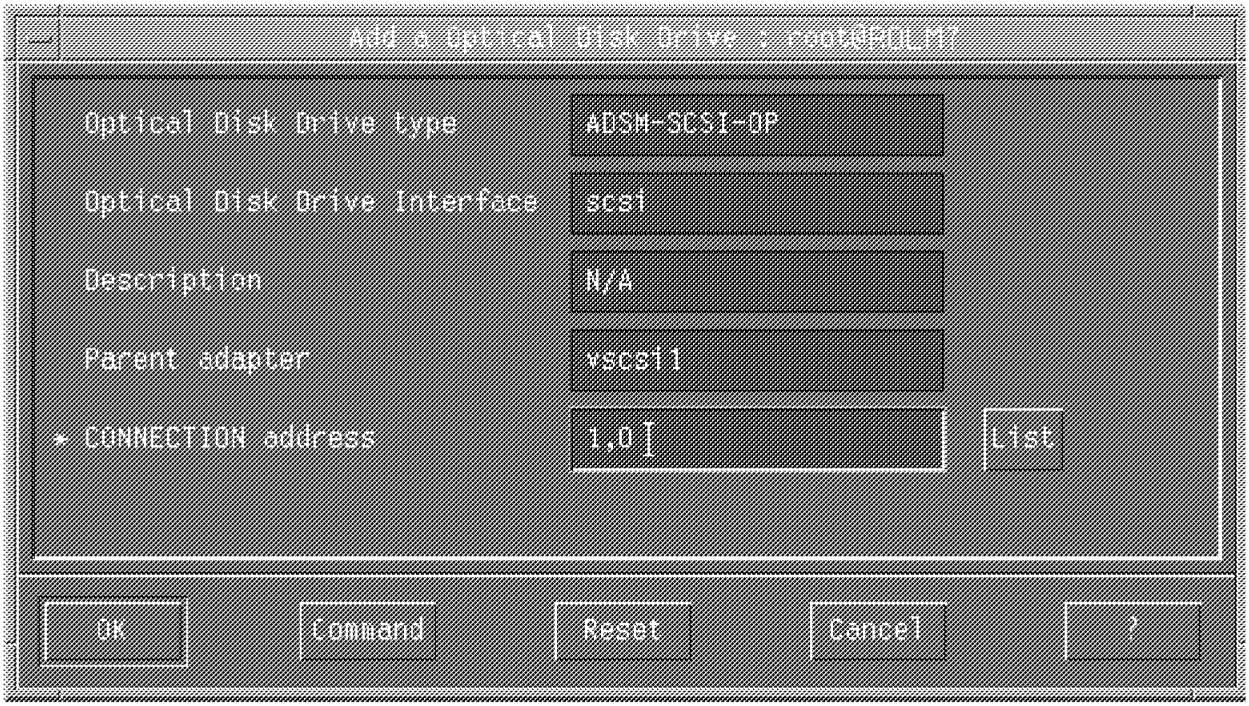
Parent Adapter

```
scsi0  Available 00-00-0S Standard SCSI I/O Controller
vscsi0 Available 00-01-00 SCSI I/O Controller Protocol Device
vscsi1 Available 00-01-01 SCSI I/O Controller Protocol Device
```

Figure 71. SCSI Card Selection

Select the adapter on which the optical drives have been detected (see “Connecting the Optical Library to the RS/6000” on page 89). The Add a

Optical Disk Drive: root@rolm7 window appears. Enter the Connection Address of the first drive, including its subaddress **1,0** (mandatory). See Figure 72 on page 98.



Add a Optical Disk Drive : root@ROLM7

Optical Disk Drive type ADSM-SCSI-OP

Optical Disk Drive Interface scsi

Description N/A

Parent adapter vscsi1

* CONNECTION address 1,0

Figure 72. Optical Drive Connection Address

Click on **OK** and wait until AIX has defined this optical drive. Its name is op0.

To define the second drive, follow the same procedure, giving it a connection address. Once defined, its name is op1.

2. Define the library.

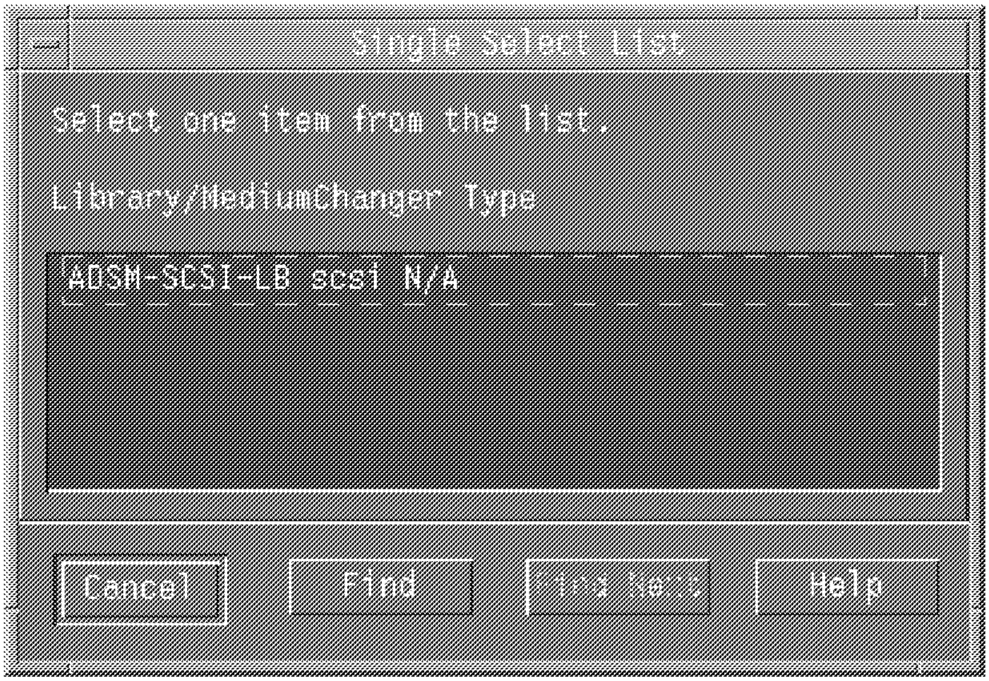
Log on to ROLM7 with the root user ID and execute the following command:

```
# smit devices
```

The System Management Interface Tool: root@rolm7 window appears, select:

```
ADSM Devices
     --> Library/MediumChanger
           --> Add a Library/MediumChanger
```

The Single Select List window appears (Figure 73 on page 99).



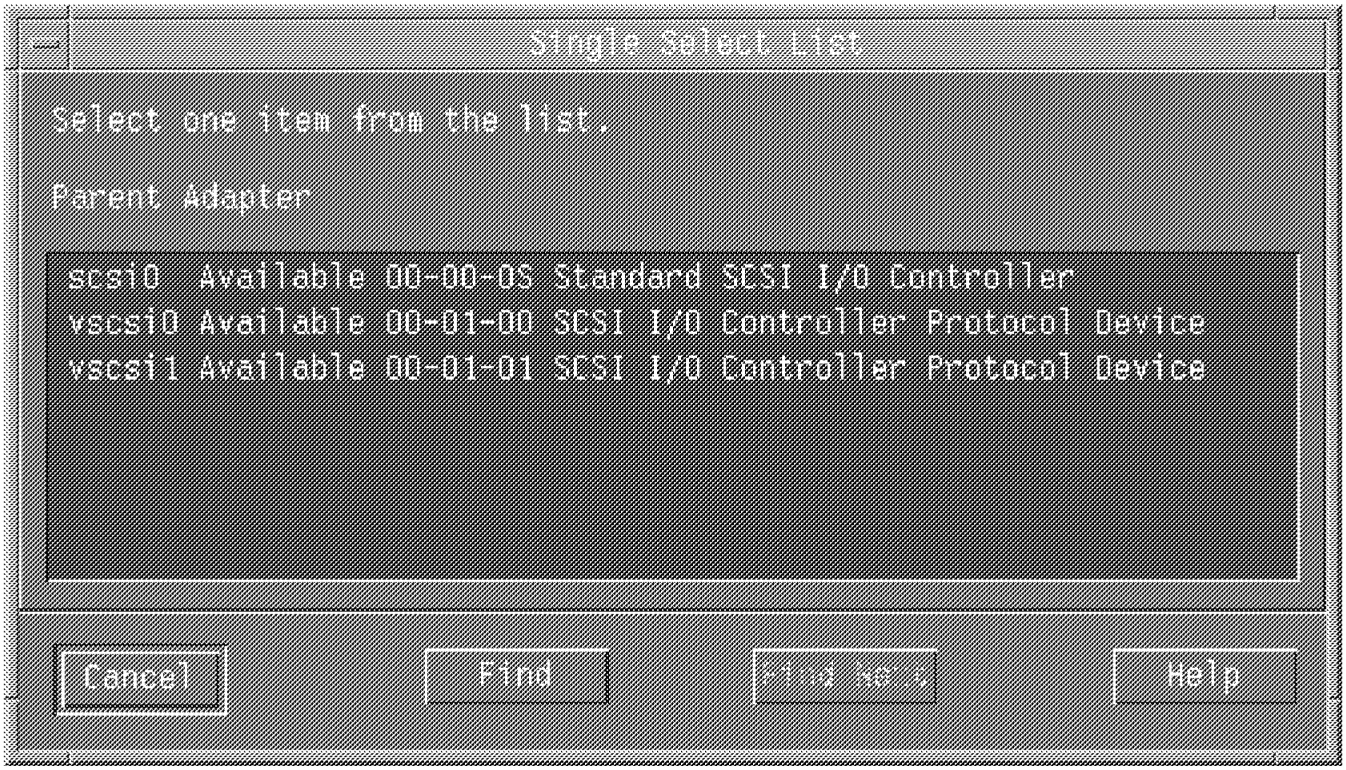
Select one item from the list.

Library/MediumChanger Type

```
ADSM-SCSI-LB scsi N/A
```

Figure 73. Library/MediumChanger Type Selection

Select the **ADSM-SCSI-LB scsi** entry. The Single Select List window appears (see Figure 74).



Select one item from the list.

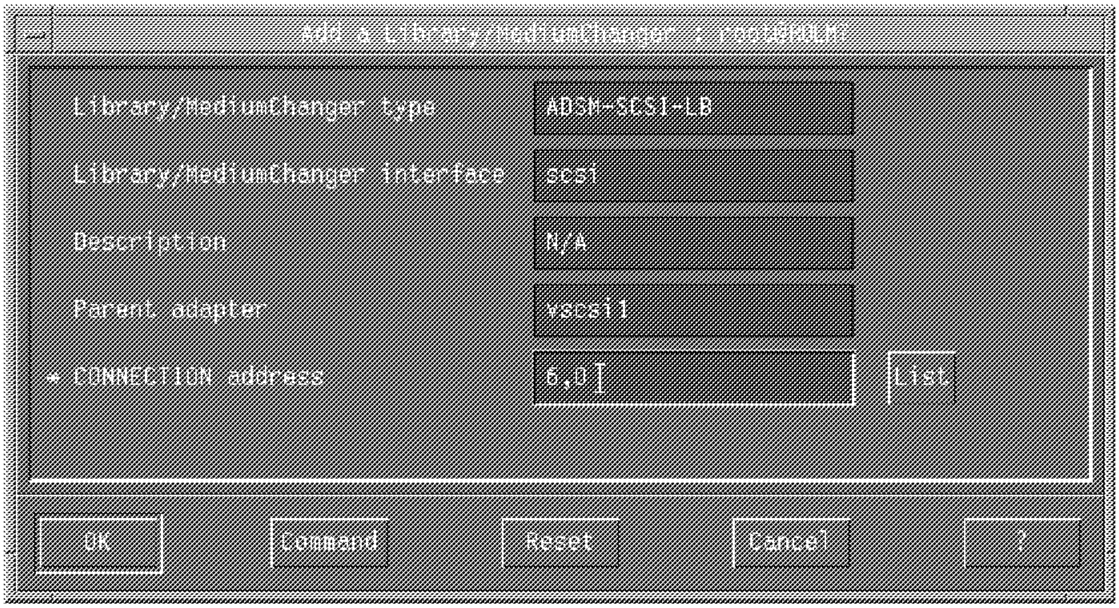
Parent Adapter

```
scsi0  Available 00-00-0S Standard SCSI I/O Controller
vscsi0 Available 00-01-00 SCSI I/O Controller Protocol Device
vscsi1 Available 00-01-01 SCSI I/O Controller Protocol Device
```

Figure 74. SCSI Card Selection

Select the adapter on which the optical drives have been detected (see“Connecting the Optical Library to the RS/6000” on page 89). When the

Add a Optical Disk Drive: root@rolm7 window appears, enter the Connection Address of the library, those provided by default with the IBM 3995 (see “Defining the SCSI Address” on page 89). Also include its subaddress **6.0** (mandatory). See Figure 75.



Add a Library/MediumChanger : root@ROLM7

Library/MediumChanger type: ADSM-SCSI-LB

Library/MediumChanger interface: scsi

Description: N/A

Parent adapter: vscsi1

* CONNECTION address: 6,0

Figure 75. Library Connection Address

Click on **OK** and wait until AIX has defined this library. Its name is lb0.

Define the ADSM Resources for our Configuration: If the ADSM server is running, start an administrative client interface (# dsmadmc) where specific commands will be executed to create the requested ADSM resources for our configuration. See Table 4 on page 95.

1. Define the policy set.

   Execute the following ADSM command to create the PSE2 policy set inside policy domain STANDARD:

   ```
   adsm> define policyset STANDARD PSE2
   ```

   The output should be:

   ```
   ANR1510I Policy set PSE2 defined in policy domain STANDARD.
   ```

2. Define the management classes.

   Execute the following ADSM commands to create the MCE2ST and MCE2LT management classes inside policy set PSE2 and policy domain STANDARD:

   ```
   adsm> define mgmtclass STANDARD PSE2 MCE2ST
   ```

   ```
   adsm> define mgmtclass STANDARD PSE2 MCE2LT
   ```

   The output should be:

   ```
   ANR1520I Management class MCE2ST defined in policy domain STANDARD,
          set PSE2.
   ```

```
ANR1520I Management class MCE2LT defined in policy domain STANDARD,
         set PSE2.
```

Before activating the new policy set PSE2, you must specify a defined management class as the default inside this new policy set. To specify MCE2ST as the default management class, execute the following ADSM command:

```
adsm> assign defmgmtclass STANDARD PSE2 MCE2ST
```

The output should be:

```
ANR1538I Default management class set to MCE2ST for policy domain
         STANDARD, set PSE2.
```

3. Define the copy group.

   Execute the following ADSM commands to create the CGE2STBU copy group inside management class MCE2ST, policy set PSE2, policy domain STANDARD, and the CGE2LTBU copy group inside management class MCE2LT, policy set PSE2 and policy domain STANDARD:

   ```
   adsm> define copygroup STANDARD PSE2 MCE2ST destination=SPE2ST
   ```

   ```
   adsm> define copygroup STANDARD PSE2 MCE2LT destination=SPE2LT
   ```

   The output should be:

   ```
   ANR1555W The BACKUP copy group in management class MCE2ST specifies a
   destination that does not refer to a defined storage pool: SPE2ST.  If
   this pool does not exist when policy set PSE2 is started, clients will
   fail when using this management class to backup files to the server.
   ANR1530I Backup copy group STANDARD defined in policy domain STANDARD,
   set PSE2, management class MCE2ST.
   ```

   ```
   ANR1555W The BACKUP copy group in management class MCE2LT specifies a
   destination that does not refer to a defined storage pool: SPE2LT.  If
   this pool does not exist when policy set PSE2 is started, clients will
   fail when using this management class to backup files to the server.
   ANR1530I Backup copy group STANDARD defined in policy domain STANDARD,
   set PSE2, management class MCE2ST.
   ```

   The output is only a warning. You should create the specified storage pool before starting the new policy set PSE2.

   Because VisualInfo does not use the archive API when migrating data to ADSM, it is not necessary to create an archive copy group.

4. Define the library.

   Execute the following ADSM command to define library LB0 in ADSM:

   ```
   adsm> define library LB0 libtype=SCSI device=/dev/lb0
   ```

   Refer to “Define the Library and its Drives in AIX” on page 96 for the AIX device resource name (/dev/lb0). The parameter libtype=SCSI is mandatory when using an automatic library through a SCSI autochanger device. The output should be:

   ```
   ANR8400I Library LB0 defined.
   ```

5. Define a drive.

Execute the following ADSM commands to define both optical drives DRV0 and DRV1 in ADSM:

```
adsm> define drive LB0 DRV0 device=/dev/rop0 element=1
```

```
adsm> define drive LB0 DRV1 device=/dev/rop1 element=2
```

Refer to "Define the Library and its Drives in AIX" on page 96 for the AIX device resource names (/dev/op0 and /dev/op1). These drives are included inside the LB0 library. The parameter ELEMENT specifies the drives SCSI address; it is mandatory. The output should be:

```
  ANR8404I Drive DRV0 defined in library LB0.
```

```
ANR8404I Drive DRV1 defined in library LB0.
```

6. Define the device class.

Execute the following ADSM command to define the optical device class DCOPTI in ADSM.

```
adsm> define devclass DCOPTI devtype=optical library=LB0 mountlimit=2
```

The DEVTYPE=OPTICAL parameter specifies that the device class uses two-sided 5.25-inch rewritable optical media.

The MOUNTLIMIT=2 parameter specifies that two platters could be mounted simultaneously, which fits the library configuration (two drives). The output should be:

```
ANR2203I Device class DCOPTI defined.
```

7. Define a storage pool.

Execute the following ADSM commands to define storage pools SPE2ST and SPE2LT, using the device class DCOPTI, in ADSM:

```
adsm> define stgpool SPE2ST DCOPTI maxscratch=10
```

```
adsm> define stgpool SPE2LT DCOPTI maxscratch=10
```

The MAXSCRATCH=10 parameter specifies that no more than 10 scratch platters could be allocated by the storage pool. This prevents any storage pool from using all the defined scratch platters. The output should be:

```
ANR2200I Storage pool SPE2ST defined (device class DCOPTI).
```

```
ANR2200I Storage pool SPE2LT defined (device class DCOPTI).
```

8. Define volumes.

In our configuration, we assign volume VI001A to storage pool SPE2ST and volume VA002A to storage pool SPE2LT. Execute the following ADSM commands to assign the volumes to their storage pools:

```
adsm> define volume SPE2ST VI001A
```

```
adsm> define volume SPE2LT VI002A
```

The output should be:

```
ANR2206I Volume VI001A defined in storage pool SPE2ST (device class
         DCOPTI).
```

```
ANR2206I Volume VI002A defined in storage pool SPE2LT (device class
         DCOPTI).
```

Volumes VI003A and VI004A are defined as scratch volumes, so we do not have to register them in any storage pool.

Before using any new resources, the policy set in which they are stored must be activated. Execute the following ADSM commands to activate the policy set PSE2:

```
adsm> activate policyset STANDARD PSE2
```

The output should be:

```
ANR1550W Management class STANDARD is NOT defined in policy set PSE2
but IS defined in the ACTIVE policy set for domain STANDARD:
files bound to this management class will be RE-BOUND to the default
management class if/when this set is started.
ANR1554W DEFAULT Management class MCE2ST in policy set STANDARD
PSE2 does not have an ARCHIVE copygroup: files will not be archived by
default if this set is activated.

Do you wish to proceed?
```

This is only a warning. As mentioned previously, we have not defined any archive copy group. So you can answer **Y** to proceed with the activation of the new policy set. The output should be:

```
ANR1514I Policy set PSE2 activated in policy domain STANDARD.
```

This completes the definition of the ADSM resources necessary for our configuration. To store data in the library, platters should be installed.

Label the Platters: Stick labels have been applied on platters (see “Applying a Label to the Optical Disk Platters” on page 89). Now we must record the labels inside the platters and format the platters. Use the AIX DSMLABEL utility.

> **Important**
>
> It is possible to encounter an error when this utility is processing and the ADSM server is running simultaneously. To avoid this error, stop ADSM when the AIX DSMLABEL is executing.
>
> When the AIX DSMLABEL utility is running, it asks for the drive to use for registering the platter label. The name of the drive to provide is the raw device name. For our configuration, these names were /dev/rop0 and /dev/rop1 (instead of /dev/op0 and /dev/op1).

To label and format the first platter VI001A, execute the following AIX command:

```
# dsmlabel -drive=/dev/rop0,1 -library=/dev/lb0 -overwrite -format
```

The output should be:

```
ANR9739I Initializing library device '/dev/lb0'.
ANR9732I Insert a new volume in the entry/exit port of library
         '/dev/lb0' with side A facing up, then enter the name (1-32
         characters) to be used for its label; or just press ENTER to
         quit this program:
```

As requested, insert the platter inside the entry/exit ports. Provide the name of the platter (VI001A) and press Enter. The output should be:

```
ANR9743I Attempting to label volume 'VI001A' using drive '/dev/rop0'...
ANR9720I Volume 'VI001A' was labeled successfully using drive '/dev/rop0'

ANR9738I Remove the volume from the entry/exit port of library
         '/dev/lb0', then press ENTER.
```

Once this operation completes successfully, the labeled platter is returned. You then label the other three platters.

It is mandatory to use the raw device drive for the device entry as well as to specify the SCSI address of this device drive; otherwise this command fails.

We used the overwrite option to overwrite an existing label. If this step is omitted, any platter with a preexisting label is rejected.

> **Important**
>
> During our test we used the -keep option to label a platter. Using this option, the label/format request completed successfully and the platter stays, as requested, inside the library. The only way to check in this preinstalled platter is to use the option search=yes with the ADSM checkin command.

Registering the Platters in the Library: ADSM is now ready to store any data for backup in the library. You should still install newly labelled platters inside the library and register them in any storage pool or as scratch.

If platters are registered in a defined storage pool, only that resource will be allowed to use them.

If platters are defined as scratch, the first storage pool that runs out of space on its existing platters will select the first available scratch disk and register it.

To register the first platter (VI001A), start the ADSM server, then an administrative client interface. Keep open both related windows, and execute from the administrative client interface the following command:

```
adsm> checkin libvolume LB0 VI001A status=private
```

On the administrative client interface, the following message is displayed:

```
ANS8003I Process number 2 started.
```

On the ADSM console, the following message is displayed:

```
ANR8422I CHECKIN LIBVOLUME:Operation for library LB0 started as process 2
ANR8323I 001: Insert OPTICAL volume VI001A R/W into entry/exit port
              of library LB0 within 60 minute(s); issue 'REPLY'
              along with the request ID when ready.
```

Enter on the console the requested data: reply 001 (001 is the process ID delivered by ADSM). The following messages are displayed on the console when the related task is executed:

```
ANR8335I 001: Verifying label of OPTICAL volume VI001A in drive DRV0
         (/dev/rop0).
ANR8328I 001: OPTICAL volume VI001A mounted in drive DRV0 (/dev/rop0).
ANR8427I CHECKIN LIBVOLUME for volume VI001A in library LB0 completed
         successfully.
ANR0985I Process 2 for CHECKIN LIBVOLUME running in the BACKGROUND
         completed with completion state SUCCESS at 22:35:04.
```

Platter VI001A is now ready to be used by ADSM.

Repeat this step for the other three platters, specifying the scratch option for both VI003A and VI004A.

Once checked, a platter is removed with the checkout command. To remove platter VI001A from library LB0, run the following command:

```
adsm> checkout libvolume LB0 VI001A
```

Important

If the ADSM console is not available (for example, if it is in a different location from the ADSM administrative client), then after running any ADSM command from the administrative client which requests manual intervention (insert platters), we must execute the following command:

```
adsm> query request
```

This allows us to retrieve the process to reply, and run it from the ADSM administrative client. Otherwise we must wait for one hour (the default configuration value) to proceed with the operation requested.

Backup Test: Before interfacing VisualInfo and ADSM, we use a backup utility provided with ADSM to ensure that our ADSM configuration can really store data on the target platters. On the ADSM client, start the GUI backup/restore application by executing the following command:

```
# dsm
```

The ADSM window is displayed (see Figure 76 on page 106).

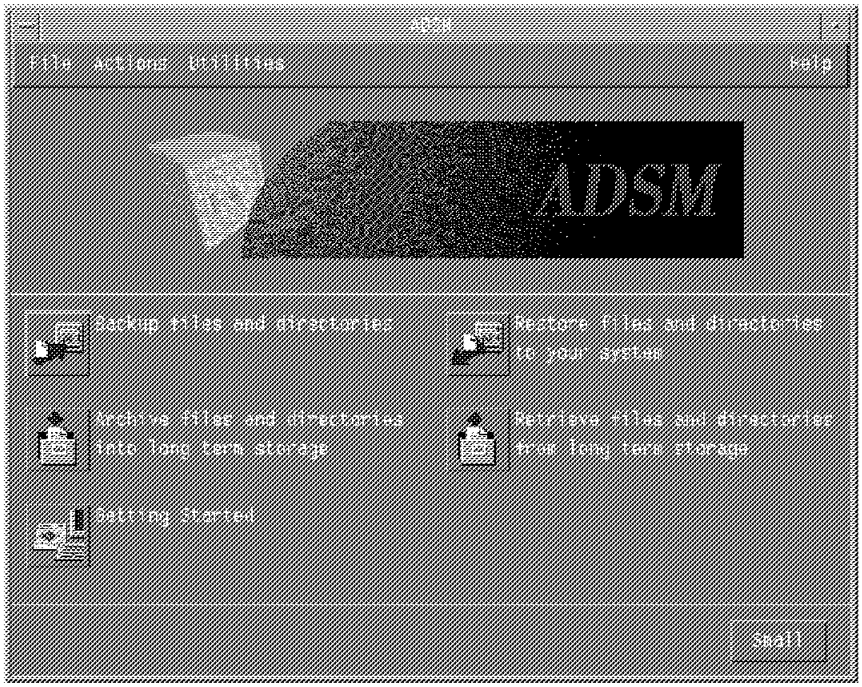
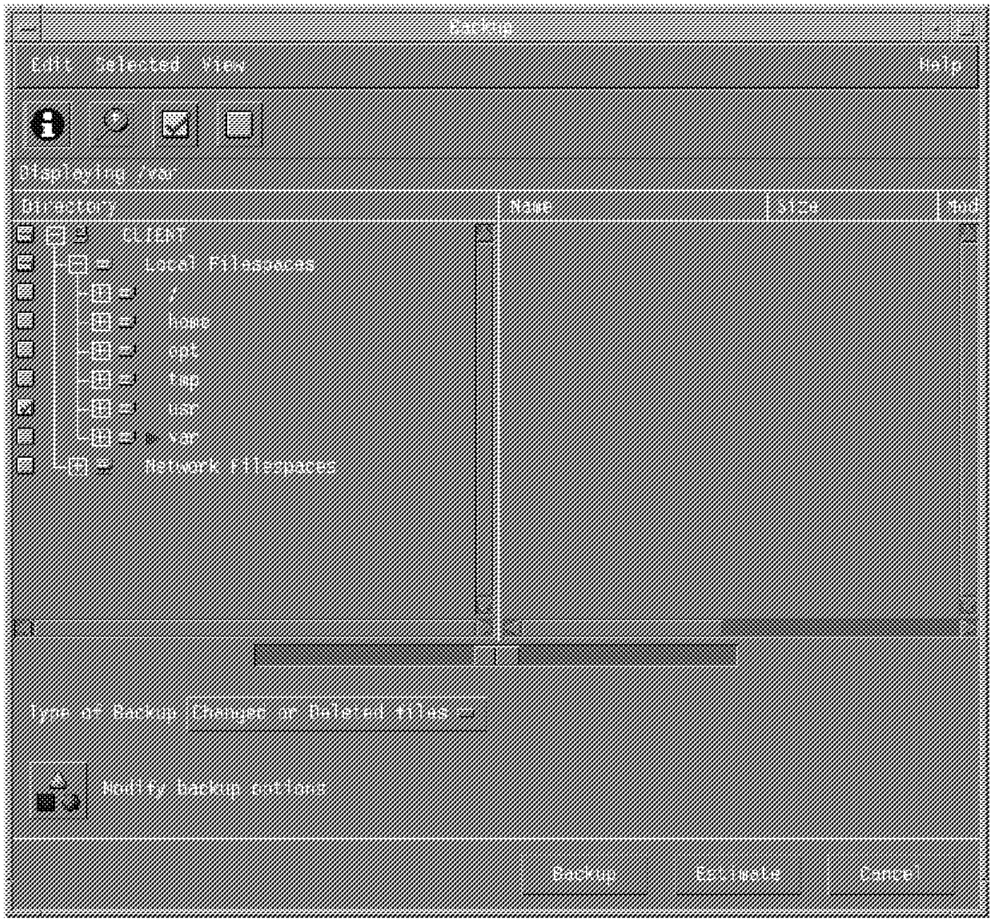

ADSM
File Actions Utilities
Help
ADSM
Backup files and directories
Restore files and directories to your system
Archive files and directories into long term storage
Retrieve files and directories from long term storage
Getting Started
Small

Figure 76. ADSM GUI Backup/Restore Application

Select the Backup files and directories option.

The ADSM file manager is displayed. Select to back up all /usr file systems (see Figure 77 on page 107).



Type of Backup: Changed or Deleted files

Figure 77. Selected Files to Back Up

Then click on **Back up** to start the backup procedure. After a while you receive the pop-up windows indicating that the backup procedure completed successfully. See Figure 78.

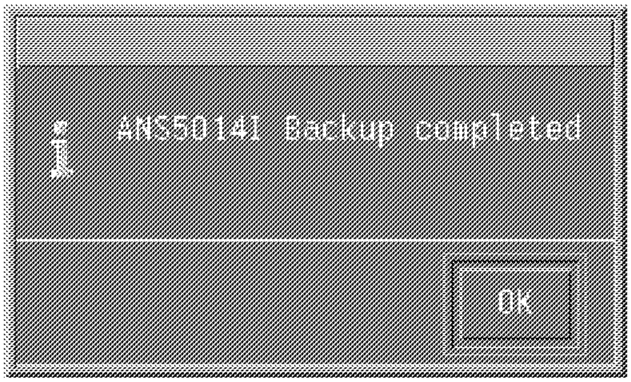

ANS5014I Backup completed

Figure 78. Backup Result

Because you have not defined a management class to use for this backup operation, ADSM selects the default management class (MCE2ST). This management class uses the SPE2ST storage pool to back up the files, and storage pool SPE2ST is defined to use volume VI001A, or a scratch volume (VI003A or VI004A) if VI001A is not available. To verify that the files are actually stored on VI001A, issue the following command from the administrative command interface:

```
adsm> query content VI0001A
```

The output lists the files registered on volume VI001A. ADSM is ready to be used by VisualInfo.

ADSM Administration: Here are some ADSM administrator′s commands to use when working with volumes:

1. Status of mounted volume

   To know which volume is mounted in one of the two drives and its related status, execute the following command:

   ```
   adsm> query mount
   ```

   In our example, the output is:

   ```
   ANR8330I OPTICAL volume VI001A is mounted R/W in drive DRV1 (/dev/rop1)
            status: IN USE.
   ```

2. Status of a volume

   To display information about a volume, execute the following command: (volume VI001A in our example)

   ```
   adsm> query volume VI001A format=detailed
   ```

   In our example, the output is:

   ```
   Volume Name: VI001A
   Storage Pool Name: SPE2ST
   Device Class Name: DCOPTI
   Estimated Capacity (MB): 2,486.3
    Pct Util: 0.6
   Volume Status: Filling
   Access: Read/Write
   Pct. Reclaimable Space: 0.0
   Scratch Volume?: No
   In Error State?: No
   Number of Writable Sides: 2
   Number of Times Mounted: 2
   Write Pass Number: 1
   Approx. Date Last Written: 02/02/98 09:50:01
   Approx. Date Last Read: 01/30/98 22:28:53
   Date Became Pending:
   Number of Write Errors: 0
   Number of Read Errors:  0
   Volume Location:
   Last Update by (administrator): ADMIN
   Last Update Date/Time: 01/30/98 22:09:42
   ```

   If you have write errors on a volume, ADSM changes the volume access status from Read/Write to ReadOnly. Once I/O errors are corrected, modify the volume access status in order to use it again. Use the following command:

   ```
   adsm> update volume VI001A access=readwrite
   ```

3. Dismount a volume

   If you need to check out a volume inserted in a drive, the `checkout` command fails because the drive is in use. Dismount the volume first. Use this command to dismount volume VI001A:

   ```
   adsm> dismount volume VI001A
   ```

6.4 Installation, Setup and Configuration of VisualInfo

This section explains how to install VisualInfo, how to set up the environment for VisualInfo components and how to configure VisualInfo for our Example 2.

6.4.1 Installation of VisualInfo

In this section, we cover installation of VisualInfo components on AIX. We recommend that you install VisualInfo from the AIX installation TaskGuide. To use the TaskGuide, you must install JDK 1.1.2, as the TaskGuide uses the Java run-time environment for installation.

6.4.1.1 Pre-Installation Tasks for VisualInfo for AIX

You must carry out the following tasks before you install VisualInfo.

1. **DB2 Instance Creation and Environment Setup**

   The library server and the object server use a DB2 instance. Complete the following steps to create DB2 instance vidb2in.

   a. Log in to ROLM3 as a root user.

   b. Once the instance is created, system administration authority group db2admin will be defined. Run the following command to create this group on AIX:

   ```
   # mkgroup db2admin
   ```

   c. Use SMIT to create user vidb2in, who will be the DB2 instance vidb2in owner, and include this user in the following groups:

   - db2admin as its primary group
   - db2admin as its group set

   d. Create the DB2 instance vidb2in by running the following command:

   ```
   #  /usr/lpp/db2_02_01/instance/db2icrt vidb2in
   ```

   Note: If you have DB2 V5 (UDB), use this command:

   ```
   #  /usr/lpp/db2_05_00/instance/db2icrt -u inst1 inst1
   ```

   e. Make sure symbolic links to the DB2 files exist by running the following command:

   ```
   # /usr/lpp/db2_02_01/cfg/db2ln
   ```

   Note: If you have DB2 V5 (UDB), use this command:

   ```
   #  /usr/lpp/db2_05_00/cfg/db2ln
   ```

   f. Insert the following lines in the vidb2in user profile (file /home/vidb2in/.profile) to set the DB2 operating environment:

   ```
   export DB2INSTANCE=vidb2in
    . /home/vidb2in/sqllib/db2profile
   ```

2. **Users Creation**

   One administrative user should be created for each VisualInfo component installed. For our example, we have to create the following users:

   - For the library server: viadmin
   - For the object server: viadmin2
   - For the SMS server: viadmin3

The SMS server is installed when you install the object server.

To create these users and set up their environment variables, follow these steps:

a. Use SMIT to create user viadmin, viadmin2 and viadmin3. Include these users in the following groups:

   - db2admin as their primary group
   - db2admin as their group set

b. Insert in their profiles

   - /home/viadmin/.profile for user viadmin
   - /home/viadmin2/.profile for user viadmin2
   - /home/viadmin3/.profile for user viadmin3

   the following lines to set the DB2 operating environment:

```
export DB2INSTANCE=vidb2in
 . /home/vidb2in/sqllib/db2profile
```

c. Because the object server user ID viadmin2 communicates with ADSM, its profile should also include the following lines:

```
export DSMI_CONFIG=/usr/lpp/adsm/bin/dsm.opt
export DSMI_DIR=/usr/lpp/adsm/bin
export DSMI_LOG=$FRNROOT
export DSM_CONFIG=/usr/lpp/adsm/bin/dsm.opt
export DSM_DIR=/usr/lpp/adsm/bin
export DSM_LOG=$FRNROOT
export PATH=$PATH:/usr/lpp/adsm/api/bin
```

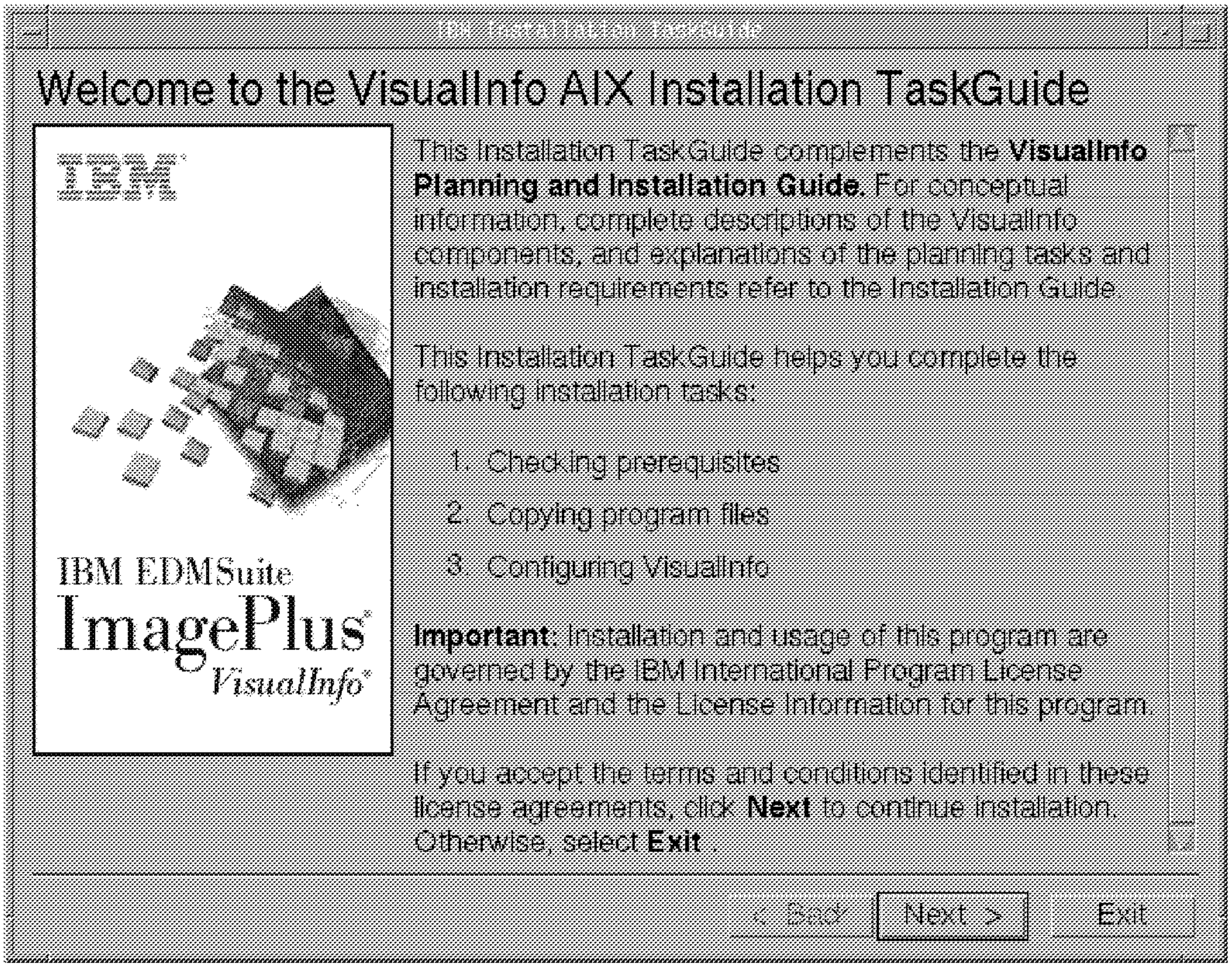
6.4.1.2 Installing VisualInfo Library Server

In this section we cover the installation of the library server for our Example 2, using the AIX installation TaskGuide.

1. Log in your system as the root user.
2. Mount the CD-ROM as you did for DB2/6000.
3. Change to the directory where the CD-ROM is mounted, that is /cdrom.
4. Now change the directory to frnaix under the /cdrom directory.
5. To begin VisualInfo for AIX installation TaskGuide, run:

```
$ ./frnxsetup.sh
```

6. You will see the screen shown in Figure 79 on page 111.



Welcome to the VisualInfo AIX Installation TaskGuide

This Installation TaskGuide complements the **VisualInfo Planning and Installation Guide.** For conceptual information, complete descriptions of the VisualInfo components, and explanations of the planning tasks and installation requirements refer to the Installation Guide.

This Installation TaskGuide helps you complete the following installation tasks:

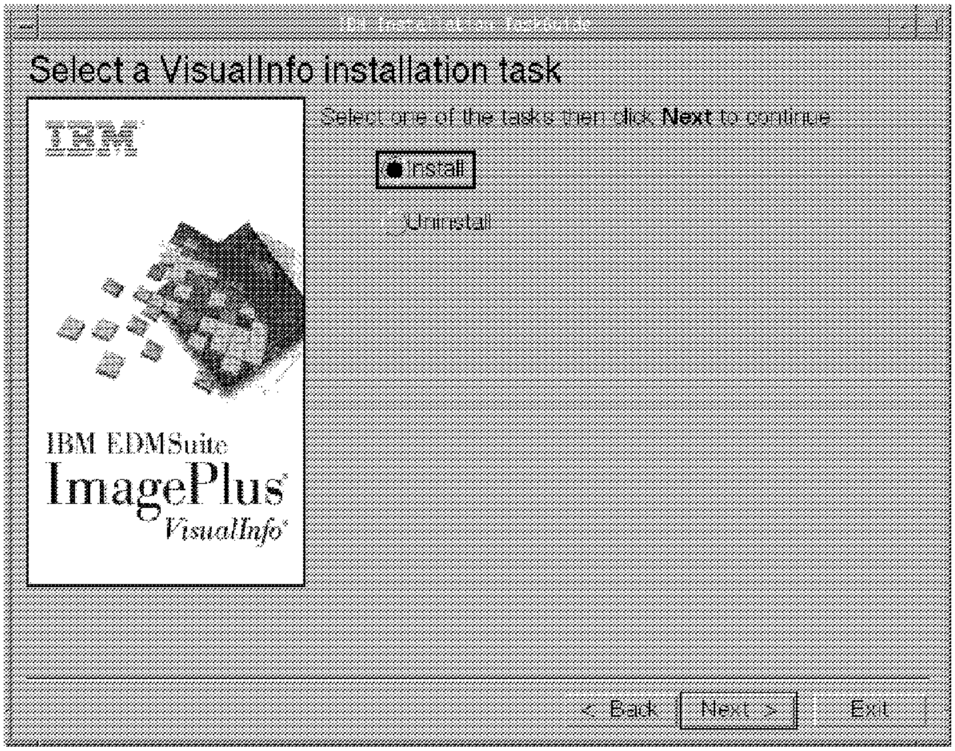
1. Checking prerequisites
2. Copying program files
3. Configuring VisualInfo

Important: Installation and usage of this program are governed by the IBM International Program License Agreement and the License Information for this program.

If you accept the terms and conditions identified in these license agreements, click **Next** to continue installation. Otherwise, select **Exit** .

Figure 79. Welcome to the VisualInfo AIX Installation TaskGuide

7. Click on **Next** to continue.
8. On the Select a VisualInfo installation task screen, shown in Figure 80 on page 112, select the **Install** option and click on **Next.**



IBM Installation TaskGuide

Select a VisualInfo installation task

Select one of the tasks then click **Next** to continue

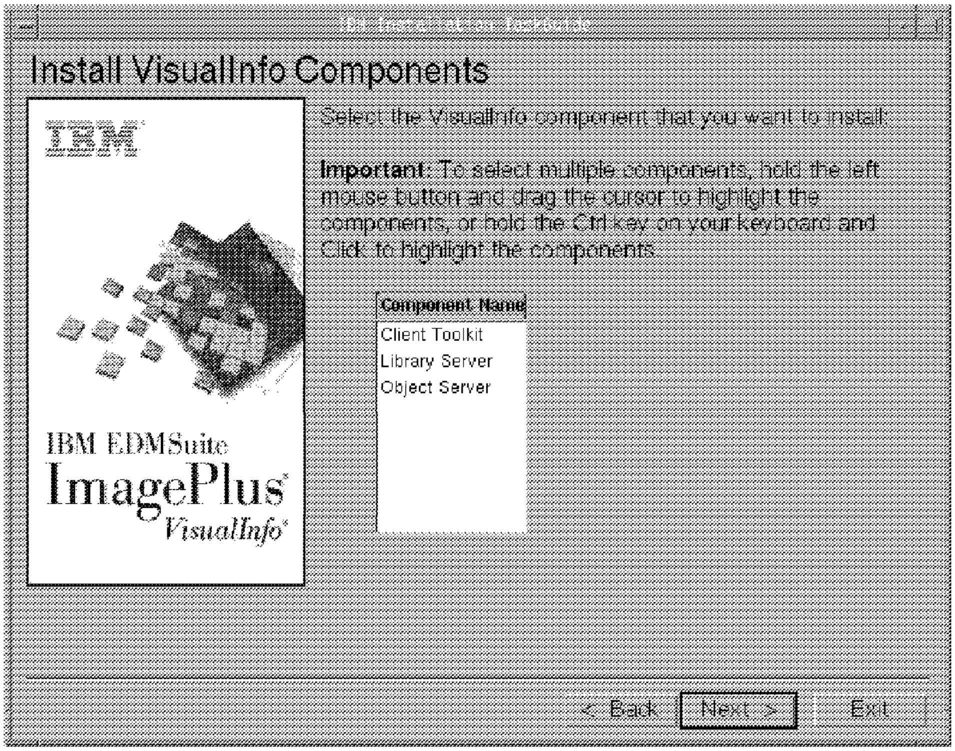
● Install

○ Uninstall

Figure 80. TaskGuide: Select a VisualInfo Installation Task

9. On the Install VisualInfo Components screen shown in Figure 81 on page 113, select **Library Server** and click on **Next**.

   You may select all three components to be installed at this time. You are prompted for the appropriate information for each component. They will be installed serially.



Install VisualInfo Components

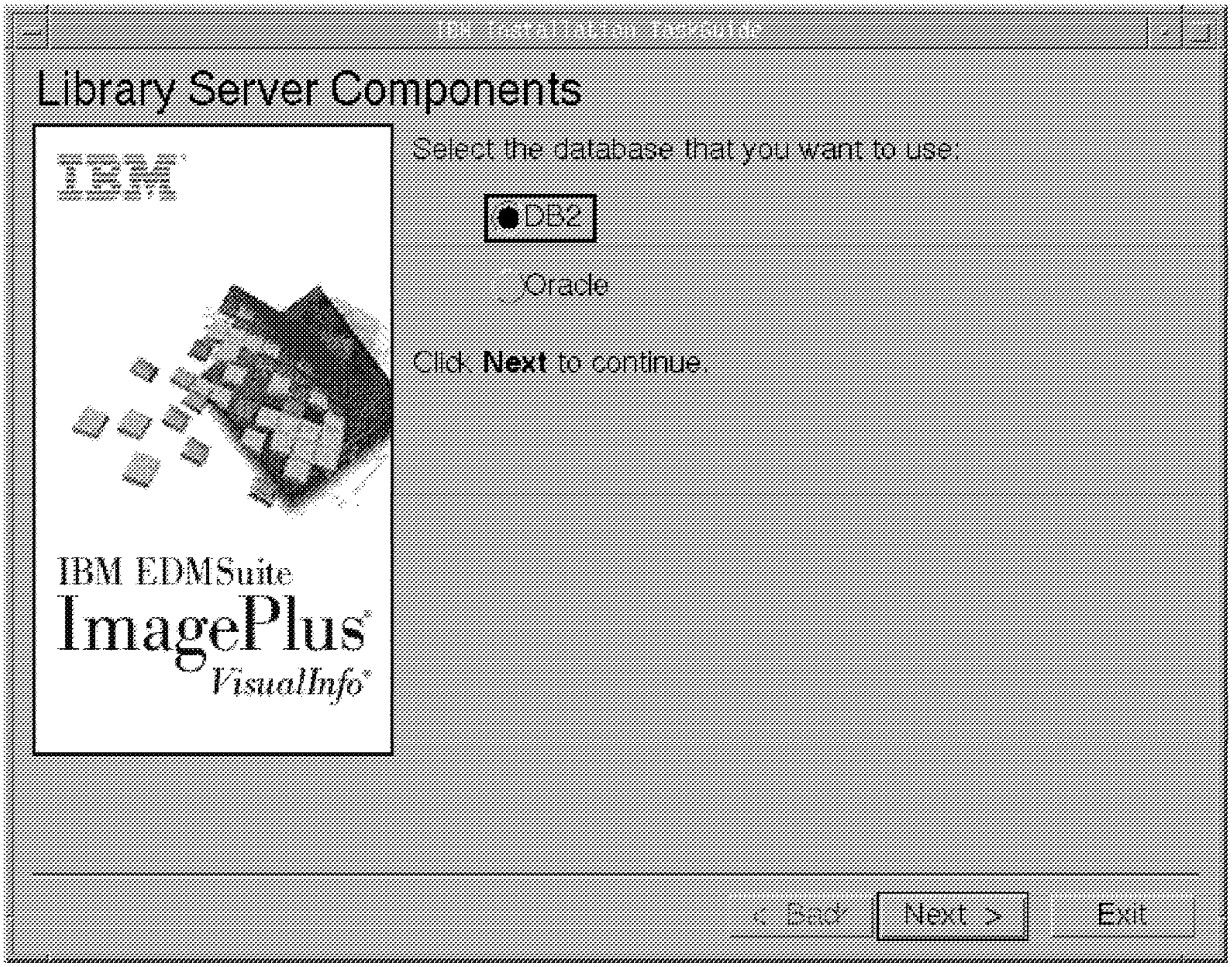
Select the VisualInfo component that you want to install:

Important: To select multiple components, hold the left mouse button and drag the cursor to highlight the components, or hold the Ctrl key on your keyboard and Click to highlight the components.

Component Name
Client Toolkit
Library Server
Object Server

Figure 81. VisualInfo TaskGuide: Install VisualInfo Components

10. On the Library Server Components screen, you are prompted to select the database you want to use. Select **DB2** or **Oracle**. We selected **DB2** for our installation, as shown in Figure 82 on page 114. Click on **Next**.



Library Server Components

Select the database that you want to use:

● DB2

○ Oracle

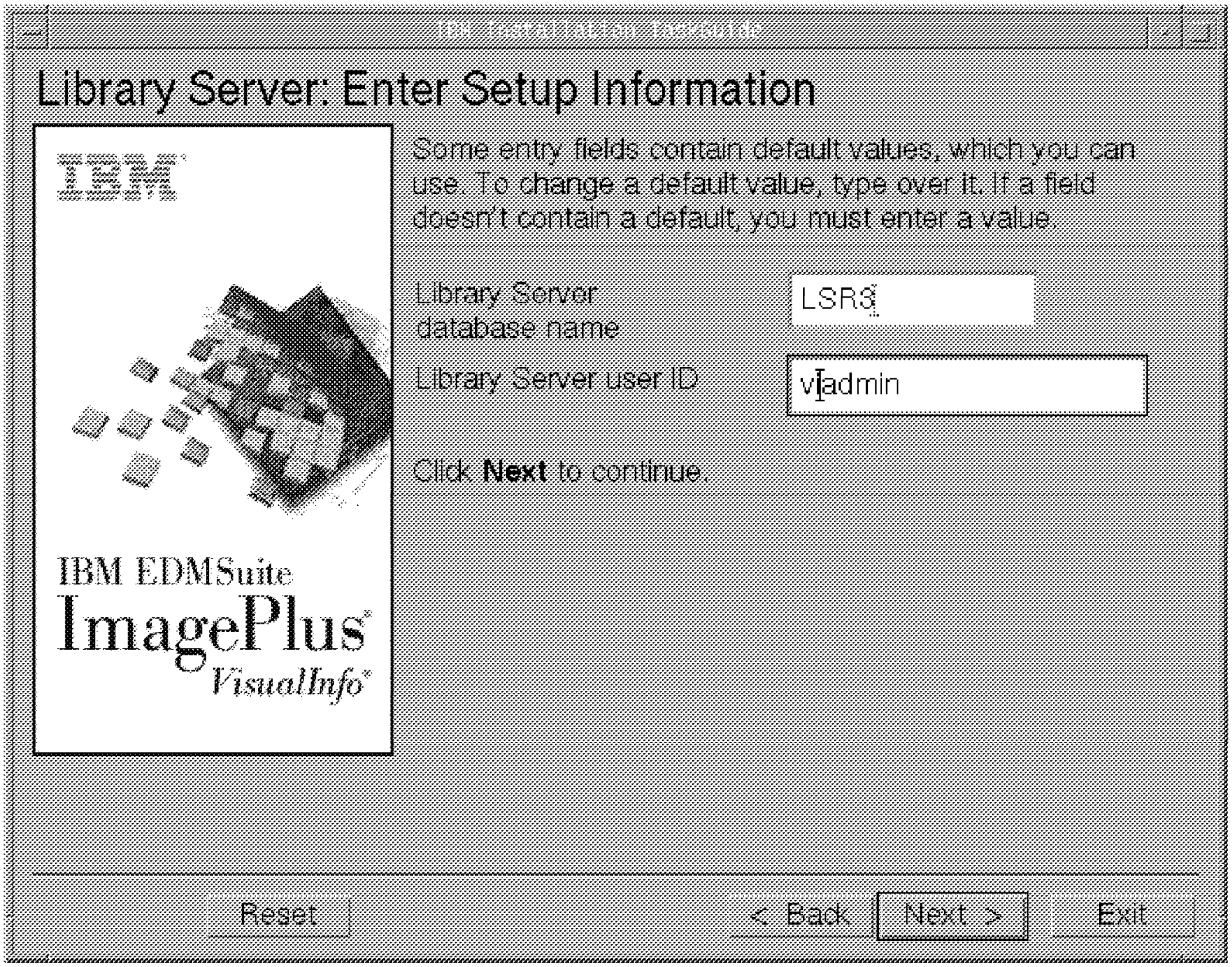
Click **Next** to continue.

Figure 82. VisualInfo TaskGuide: Library Server Components

11. On the Library Server: Enter Setup Information screen, Figure 83 on page 115, supply the following:

    - Library server database name: LSR3
    - Library server user ID: viadmin

    Then click on **Next**.



Library Server: Enter Setup Information

Some entry fields contain default values, which you can use. To change a default value, type over it. If a field doesn't contain a default, you must enter a value.

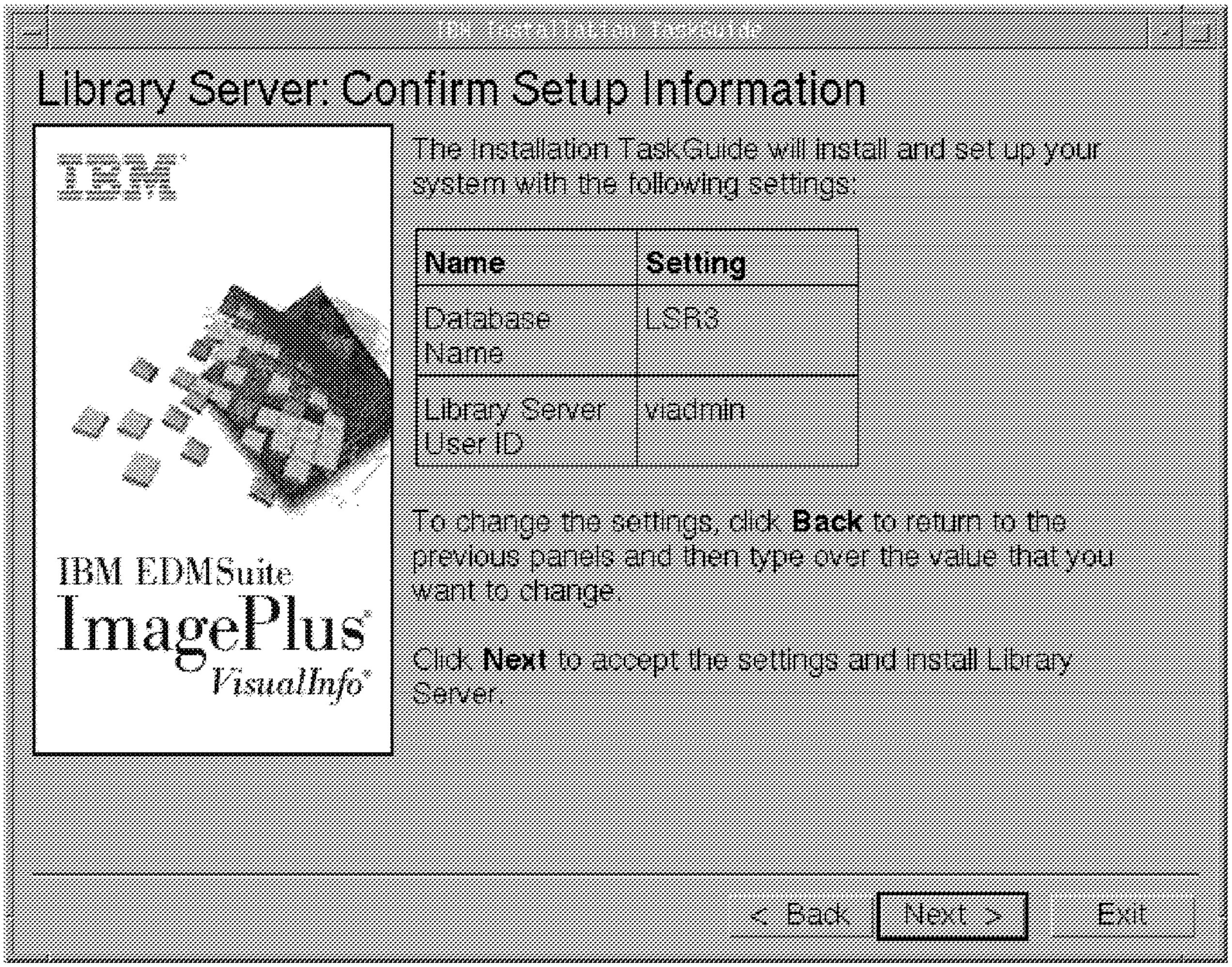
Library Server database name: LSR3

Library Server user ID: vladmin

Click **Next** to continue.

Figure 83. VisualInfo TaskGuide Library Server: Enter Setup Information

12. Figure 84 on page 116 is displayed. Confirm the accuracy of the setup information supplied by clicking on **Next**.



Library Server: Confirm Setup Information

The Installation TaskGuide will install and set up your system with the following settings:

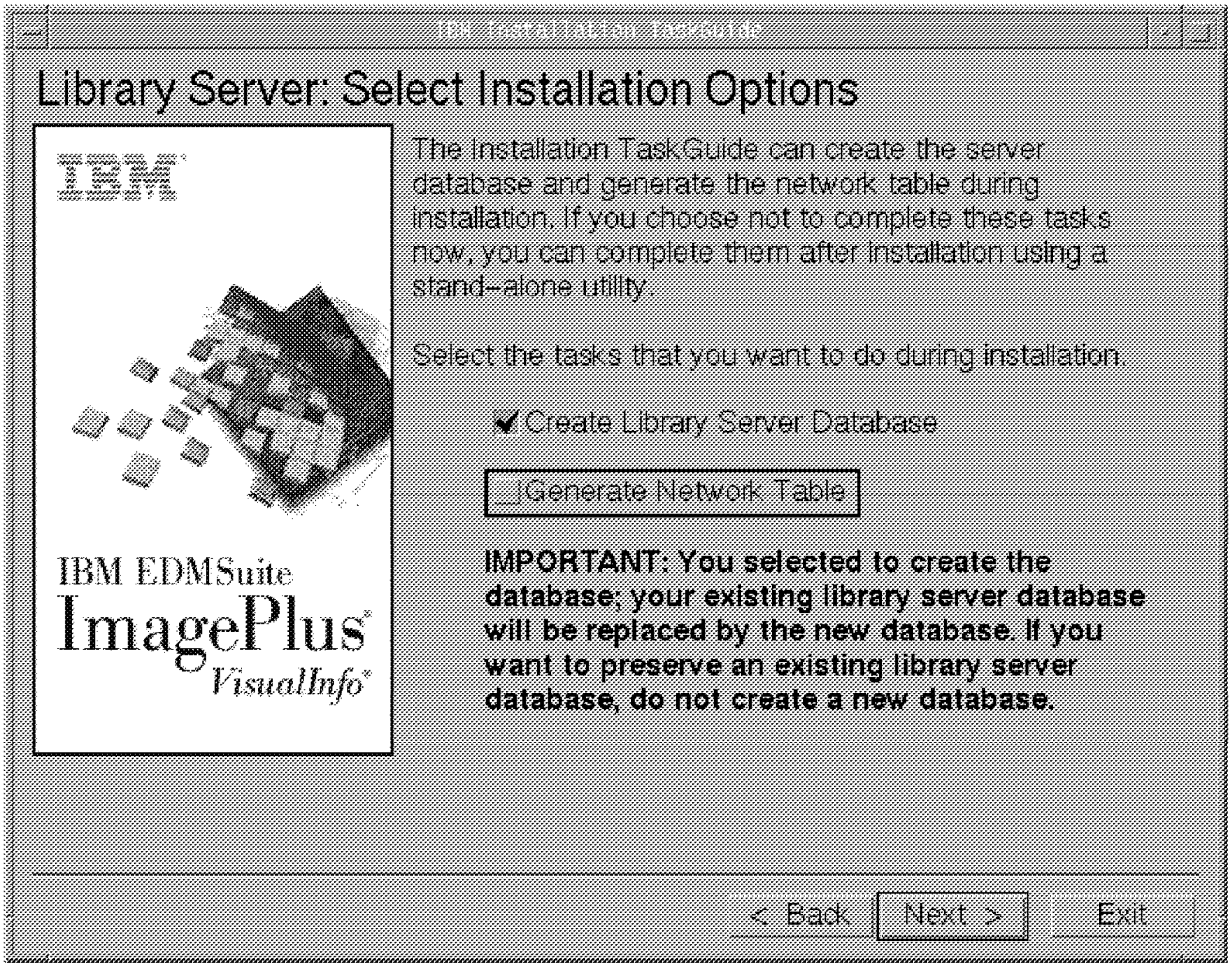
Name	Setting
Database Name	LSR3
Library Server User ID	viadmin

To change the settings, click **Back** to return to the previous panels and then type over the value that you want to change.

Click **Next** to accept the settings and install Library Server.

Figure 84. VisualInfo TaskGuide Library Server: Confirm Setup Information

13. On the Library Server: Select Installation Options screen shown in Figure 85 on page 117, we selected **Create Library server Database**. If you have gathered the network information for VisualInfo, you can generate a custom network table now. We recommend generating the custom network table *after* installation. Click on **Next**.



Library Server: Select Installation Options

The Installation TaskGuide can create the server database and generate the network table during installation. If you choose not to complete these tasks now, you can complete them after installation using a stand-alone utility.

Select the tasks that you want to do during installation.

☑ Create Library Server Database

□ Generate Network Table

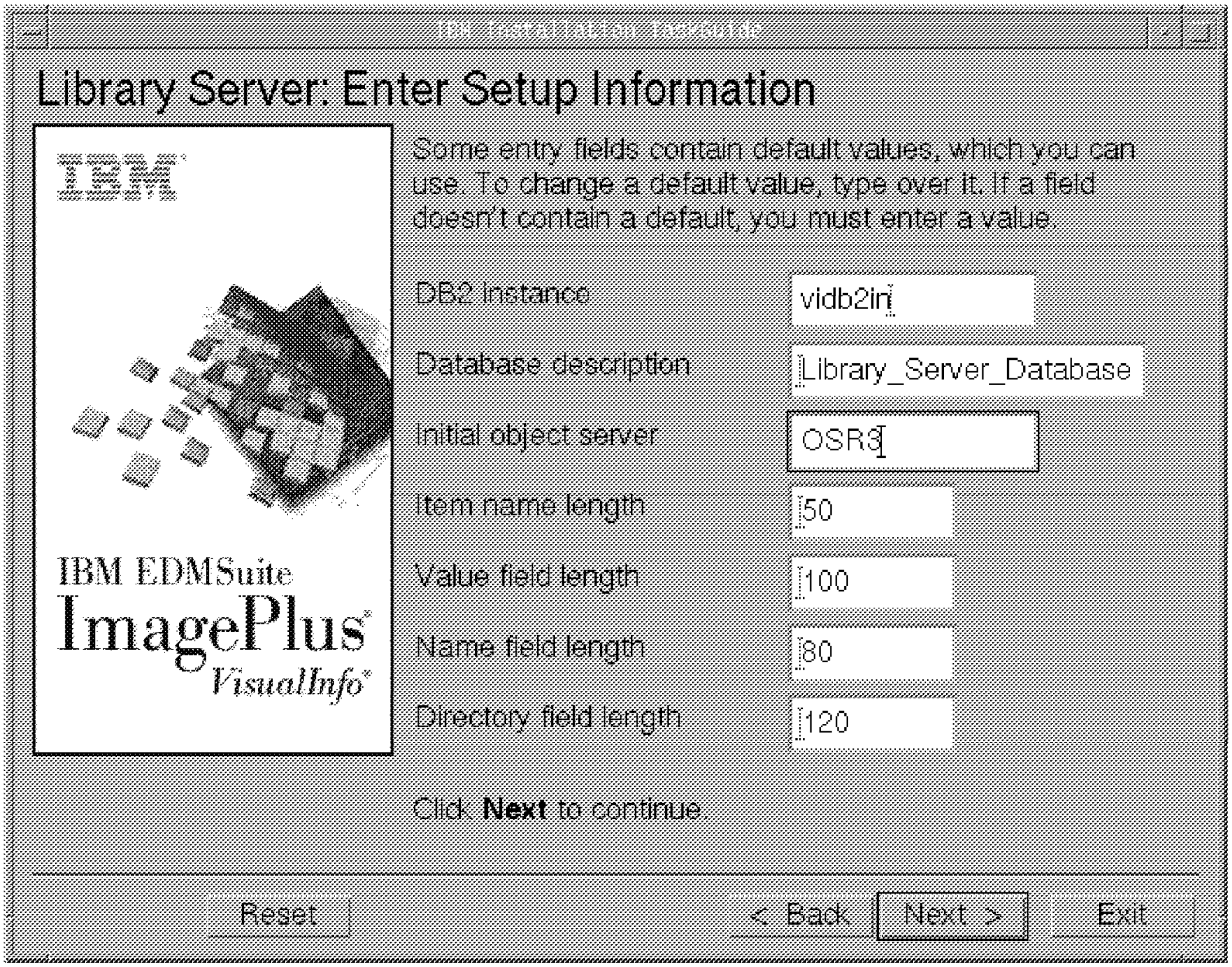
IMPORTANT: You selected to create the database; your existing library server database will be replaced by the new database. If you want to preserve an existing library server database, do not create a new database.

Figure 85. VisualInfo TaskGuide Library Server: Select Installation Options

14. On the Library Server: Enter Setup Information screen shown in Figure 86 on page 118, supply the setup information for LSR3, as follows:

    - DB2 Instance: vidb2in
    - Initial object server: OSR3

    Then click on **Next**. (We kept the default field lengths.)



Library Server: Enter Setup Information

Some entry fields contain default values, which you can use. To change a default value, type over it. If a field doesn't contain a default, you must enter a value.

DB2 instance: vidb2in

Database description: Library_Server_Database

Initial object server: OSR3

Item name length: 50

Value field length: 100

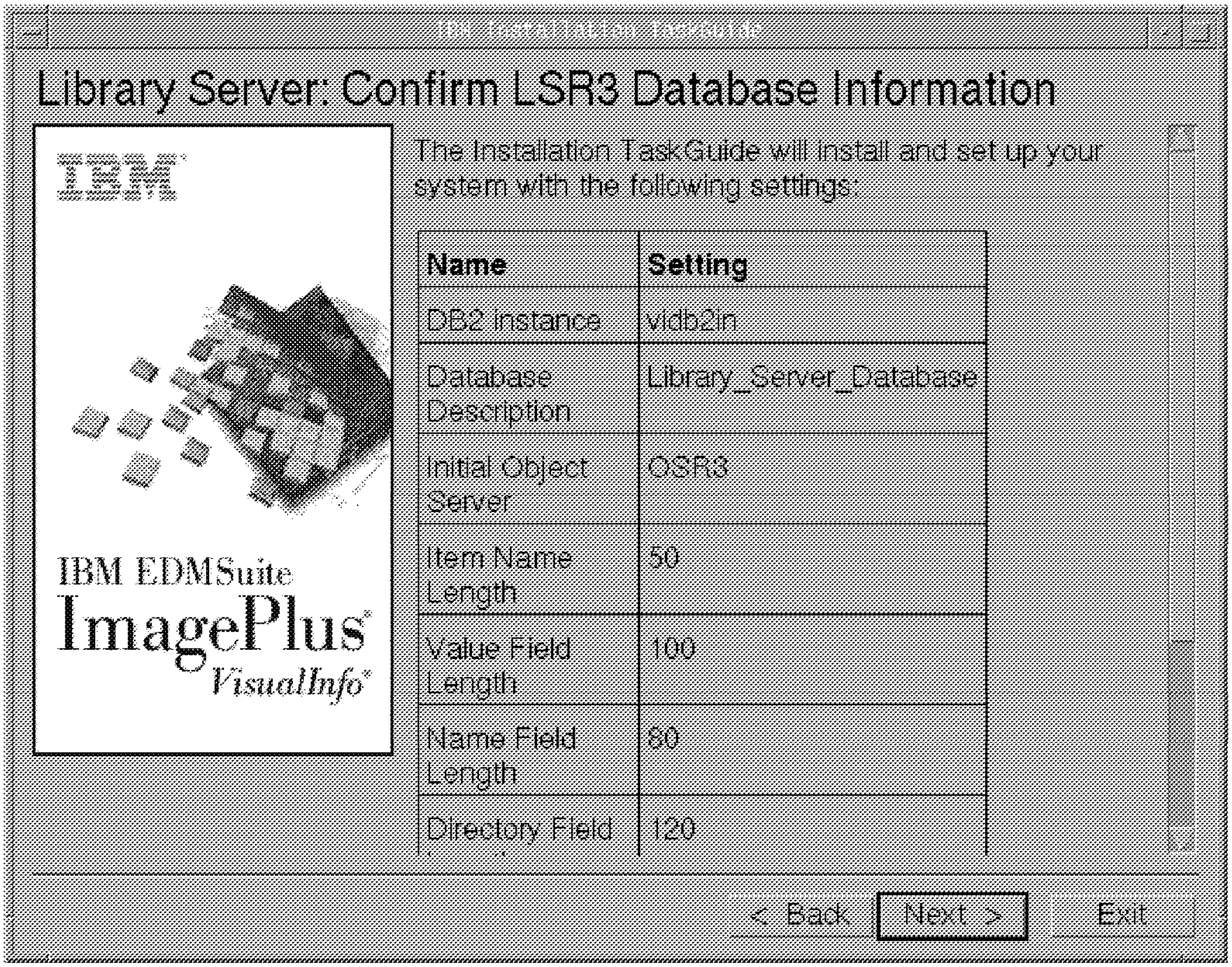
Name field length: 80

Directory field length: 120

Click **Next** to continue.

Figure 86. VisualInfo TaskGuide Library Server: Enter Setup Information

15. You can confirm the information you have supplied or click on **Back** to review the LSR3 database setup (see Figure 87 on page 119). Click on **Next** to continue.



Library Server: Confirm LSR3 Database Information

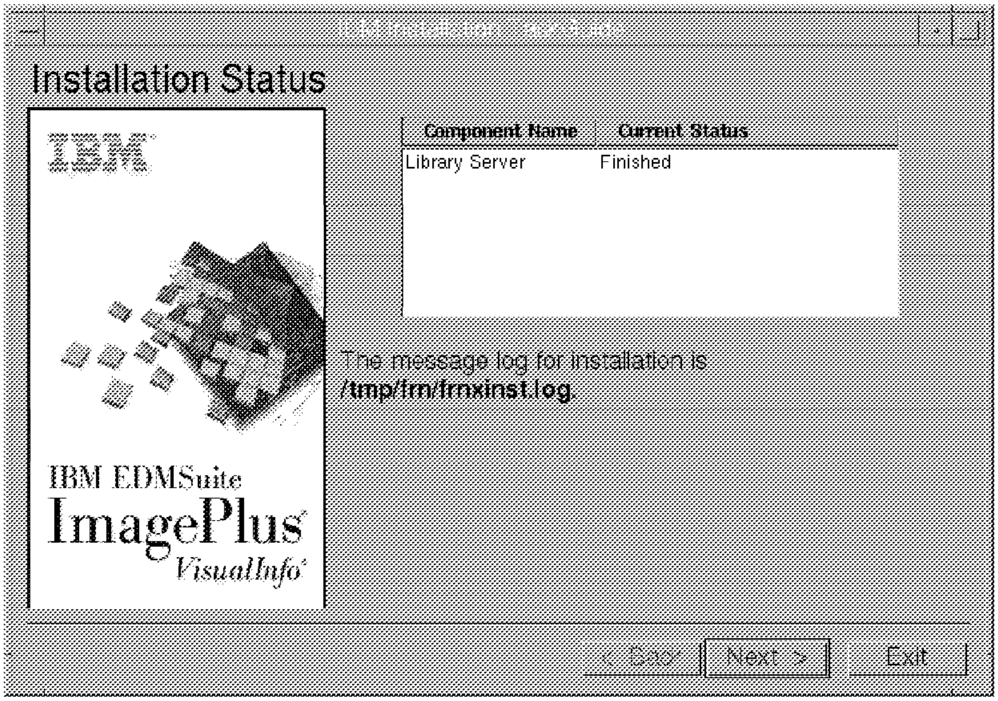
The Installation TaskGuide will install and set up your system with the following settings:

Name	Setting
DB2 instance	vidb2in
Database Description	Library_Server_Database
Initial Object Server	OSR3
Item Name Length	50
Value Field Length	100
Name Field Length	80
Directory Field	120

Figure 87. TaskGuide Library Server: Confirm LSR3 Database Information

16. The installation is successful if the Installation Status screen (Figure 88 on page 120) shows “Finished” under current status.

    If the installation was not successful, exit TaskGuide and check the /tmp/frn/frnxinst.log file to see the possible reasons for the failure.



Installation Status

Component Name	Current Status
Library Server	Finished

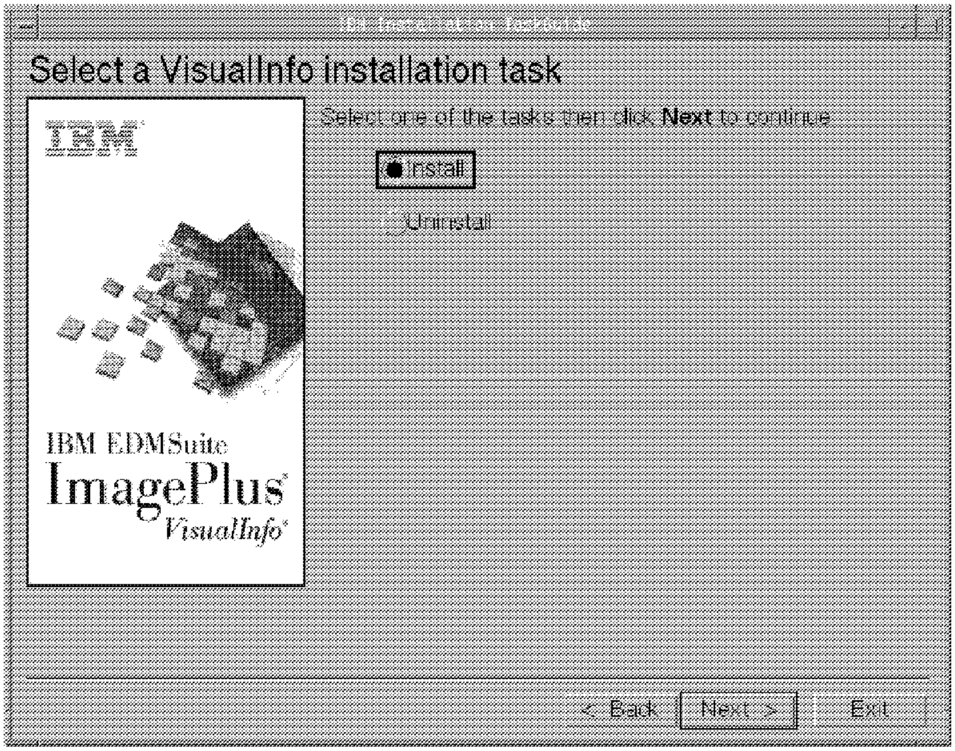
The message log for installation is **/tmp/frn/frnxinst.log.**

Figure 88. VisualInfo TaskGuide Library Server: Installation Status

6.4.1.3 Installing VisualInfo Object Server for AIX

You can install the object server after you have finished installing the library server. From this Installation Status window, click on **Next**. On the Select a VisualInfo installation task screen shown in Figure 89 on page 121, select **Install** and click on **Next**.

If you have selected multiple components, the next component is installed now.



IBM Installation TaskGuide

Select a VisualInfo installation task

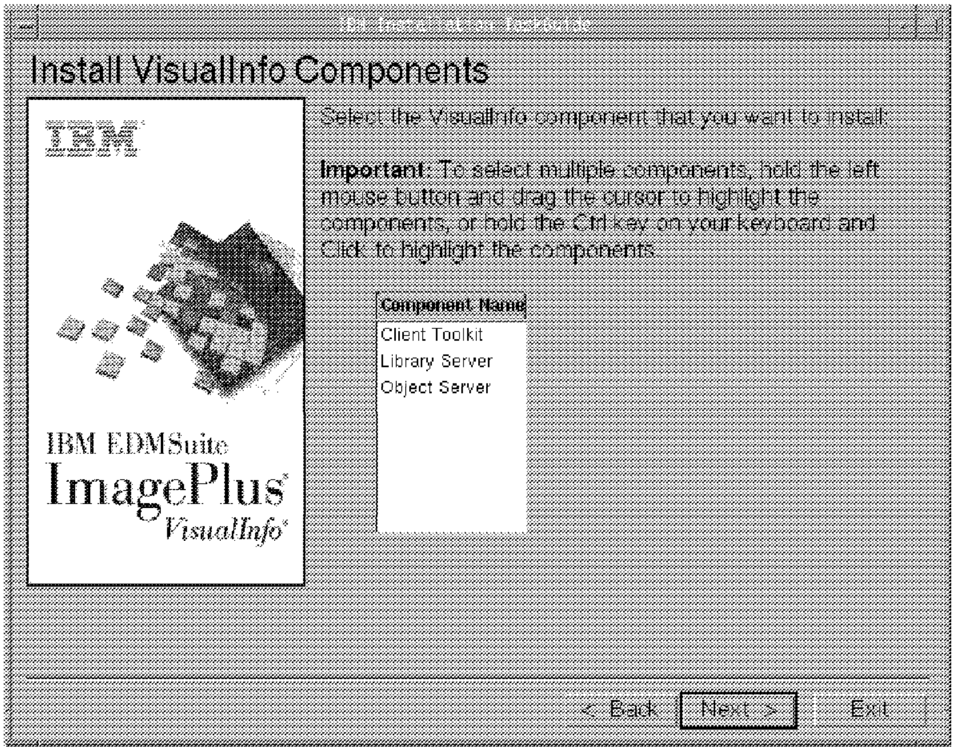
Select one of the tasks then click **Next** to continue

● Install

○ Uninstall

Figure 89. VisualInfo TaskGuide: Select a VisualInfo Installation Task

1. On the Install VisualInfo Components screen, Figure 90 on page 122, select the **Object Server** component. This also installs the SMS server. Click on **Next.**



IBM Installation TaskGuide

Install VisualInfo Components

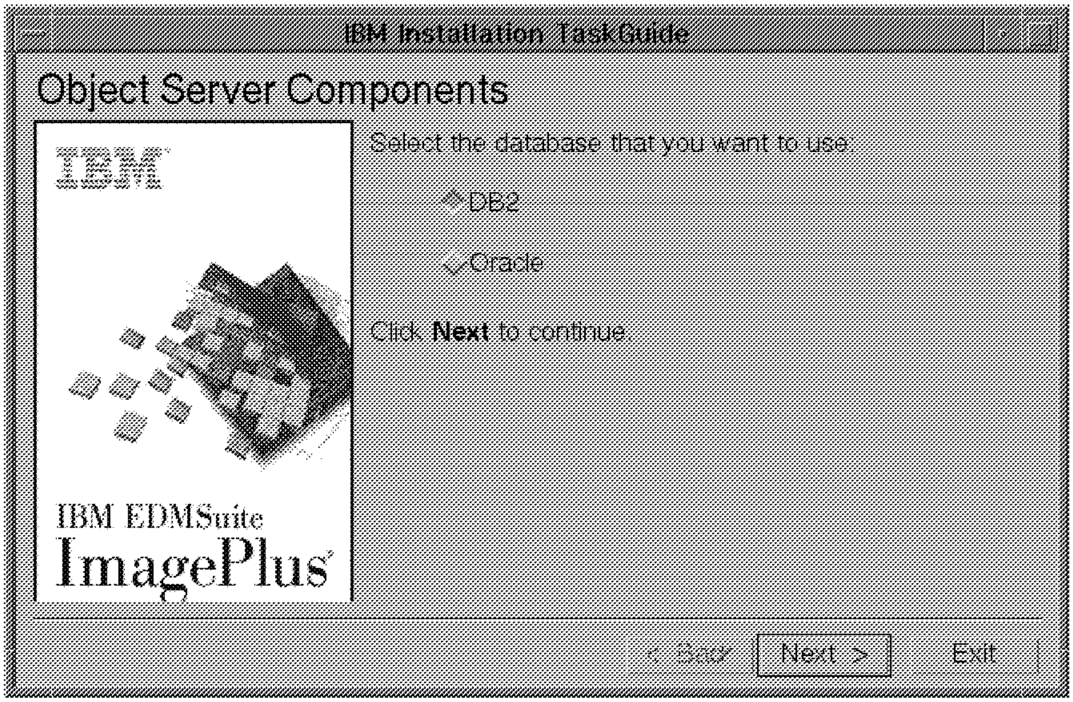
Select the VisualInfo component that you want to install:

Important: To select multiple components, hold the left mouse button and drag the cursor to highlight the components, or hold the Ctrl key on your keyboard and Click to highlight the components.

Component Name
Client Toolkit
Library Server
Object Server

Figure 90. VisualInfo TaskGuide: Select VisualInfo Components

2. On the Object server Components screen in Figure 91 on page 123, you are prompted to select the database you want to use. Select **DB2** and click on **Next.**



Object Server Components

Select the database that you want to use:

◆ DB2

◇ Oracle

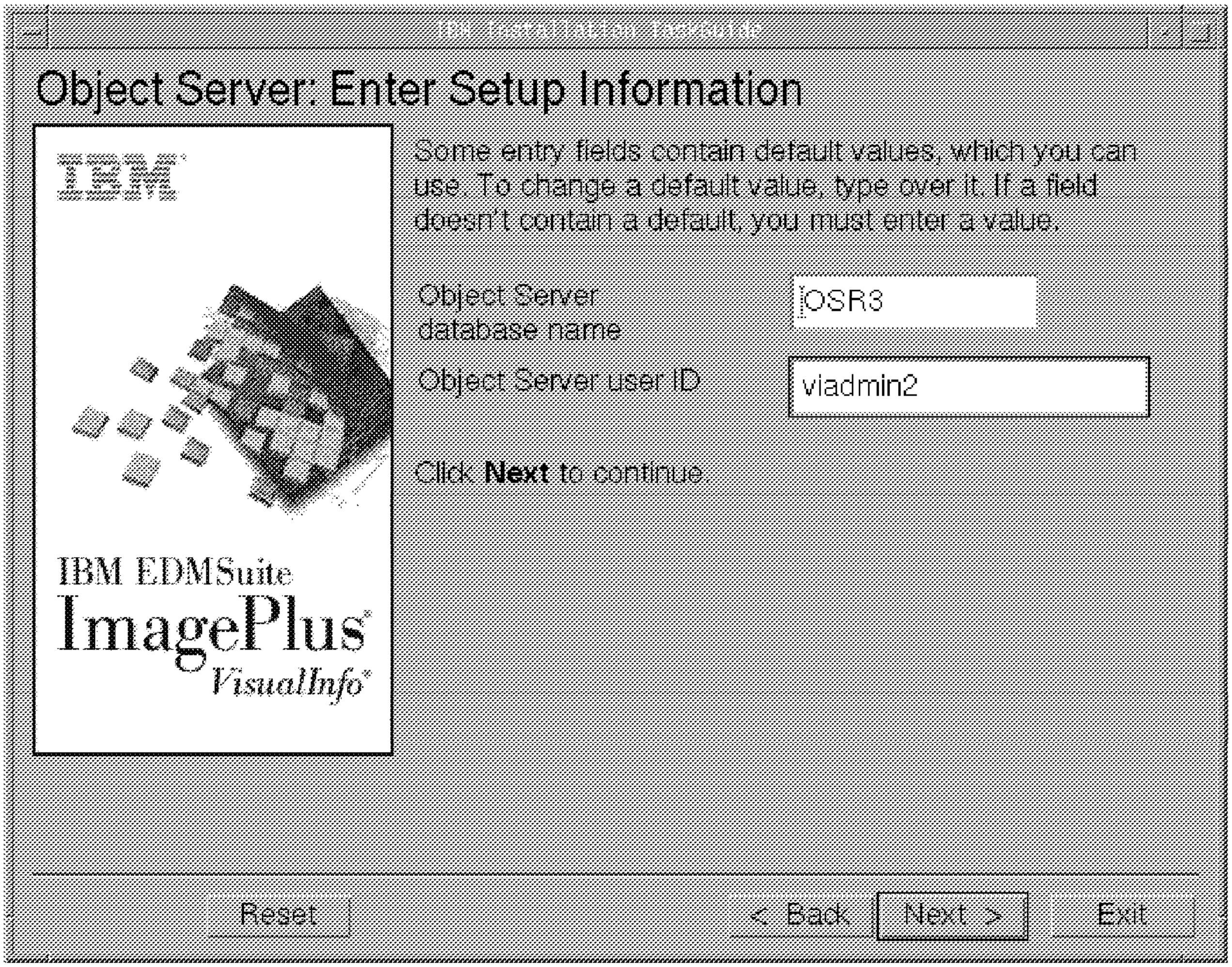
Click **Next** to continue.

Figure 91. VisualInfo TaskGuide Object Server: Object Server Components

3. On the next screen, enter the setup information as shown in Figure 92 on page 124. Supply the following:

   - The object server database name: OSR3
   - The object server user ID: viadmin2

   Then click on **Next.**



IBM Installation TaskGuide

Object Server: Enter Setup Information

Some entry fields contain default values, which you can use. To change a default value, type over it. If a field doesn't contain a default, you must enter a value.

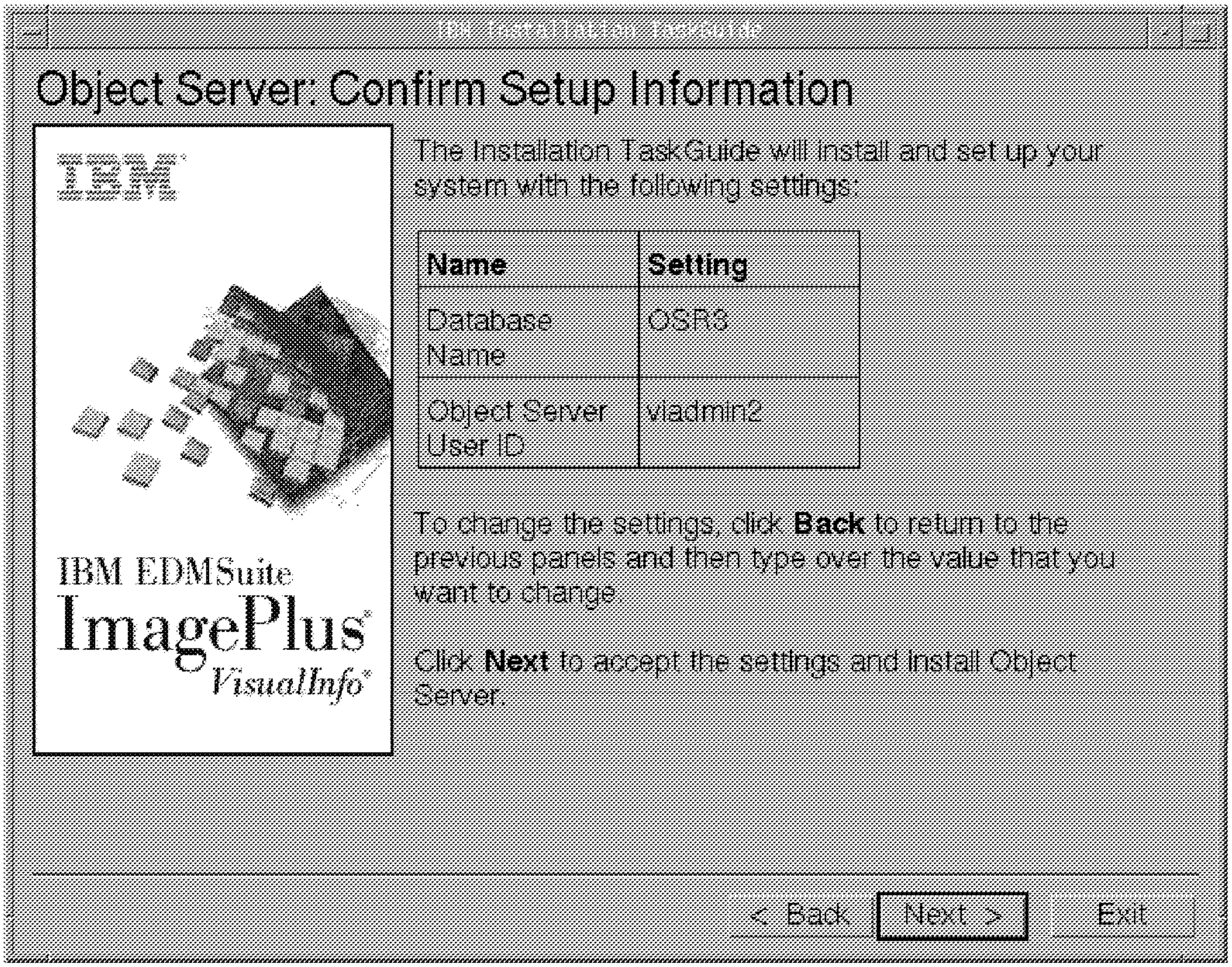
Object Server database name: OSR3

Object Server user ID: viadmin2

Click **Next** to continue.

Figure 92. VisualInfo TaskGuide Object Server: Enter Setup Information

4. On the Object server Confirm Setup Information screen, Figure 93 on page 125, verify the accuracy of the setup information you have supplied. Click on **Next** to accept the settings.



Object Server: Confirm Setup Information

The Installation TaskGuide will install and set up your system with the following settings:

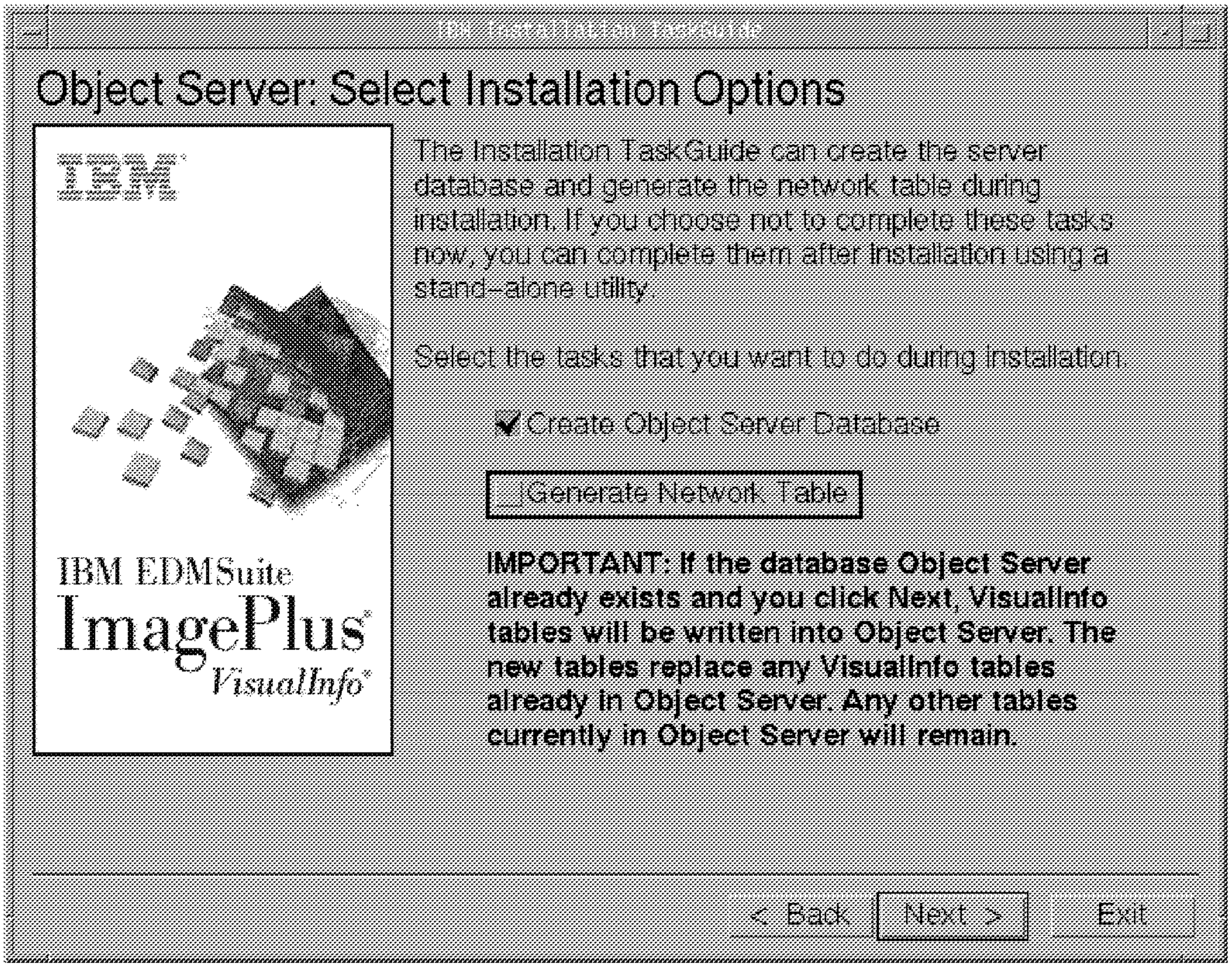
Name	Setting
Database Name	OSR3
Object Server User ID	vladmin2

To change the settings, click **Back** to return to the previous panels and then type over the value that you want to change.

Click **Next** to accept the settings and install Object Server.

Figure 93. VisualInfo TaskGuide Object Server: Confirm Setup Information

5. On the Object Server: Select Installation Options screen, Figure 94 on page 126, we selected **Create Object Server Database** only. If you have gathered the network information for VisualInfo, you can generate a custom network table at this time with TaskGuide. We recommend generating the network table *after* installation. Click on **Next.**



Object Server: Select Installation Options

The Installation TaskGuide can create the server database and generate the network table during installation. If you choose not to complete these tasks now, you can complete them after installation using a stand-alone utility.

Select the tasks that you want to do during installation.

☑ Create Object Server Database

□ Generate Network Table

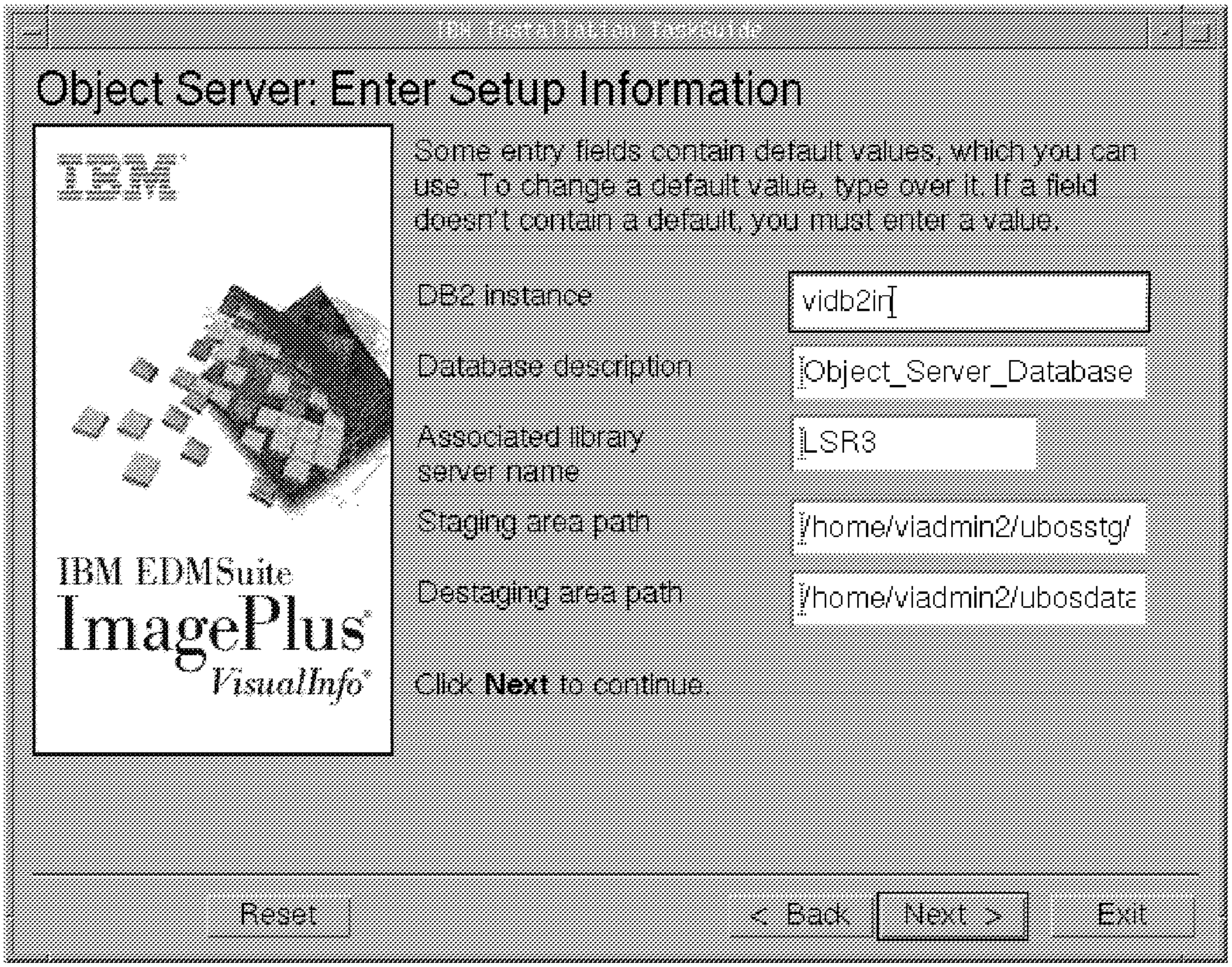
IMPORTANT: If the database Object Server already exists and you click Next, VisualInfo tables will be written into Object Server. The new tables replace any VisualInfo tables already in Object Server. Any other tables currently in Object Server will remain.

Figure 94. Object Server: Select Installation Options

6. On the Object Server: Enter Setup Information screen, Figure 95 on page 127, supply the setup information for OSR3:

   - DB2 instance: vidb2in
   - Associated library server: LSR3

   Then click on **Next**.



Object Server: Enter Setup Information

Some entry fields contain default values, which you can use. To change a default value, type over it. If a field doesn't contain a default, you must enter a value.

DB2 instance: vidb2in

Database description: Object_Server_Database

Associated library server name: LSR3

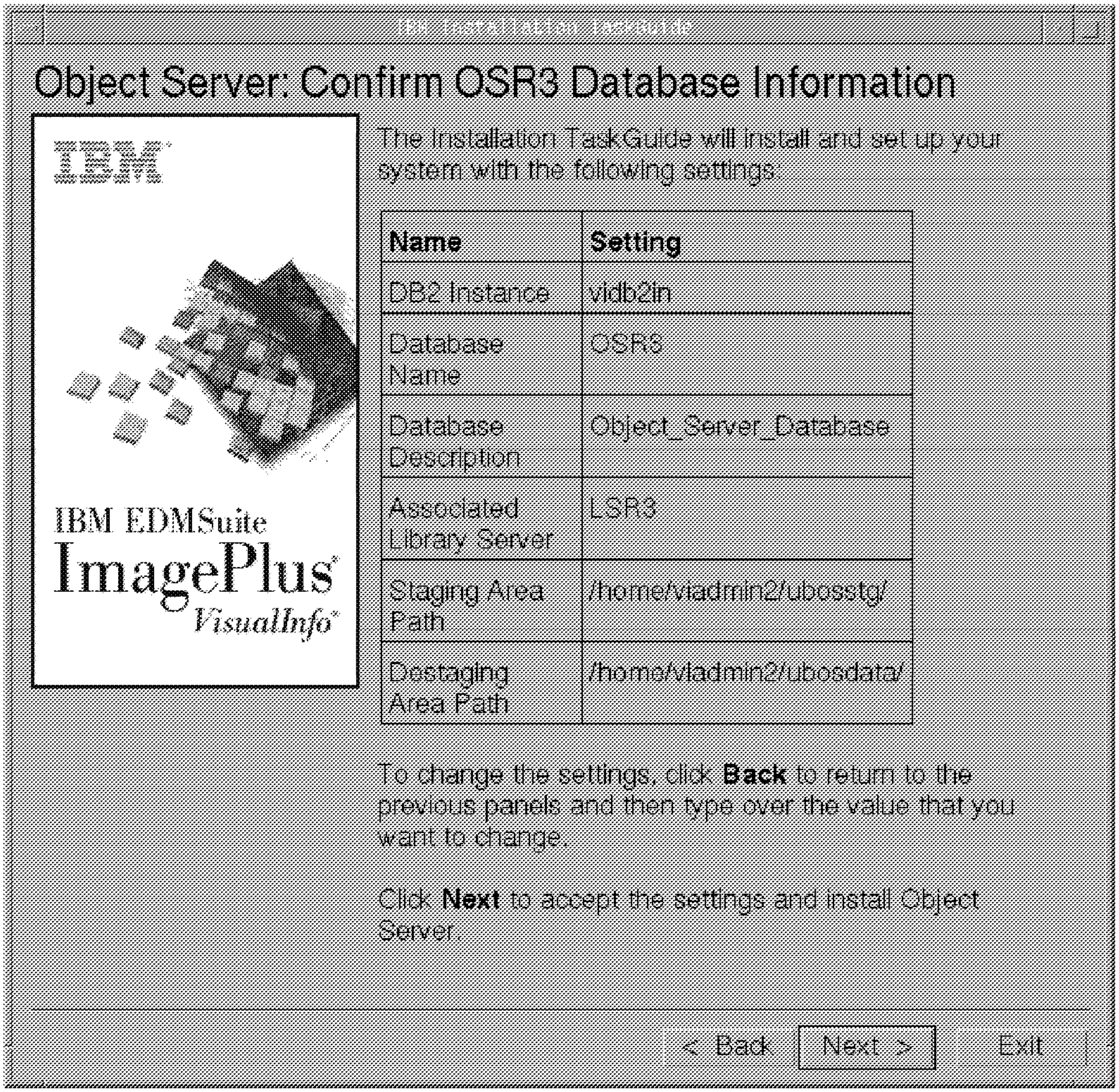
Staging area path: /home/viadmin2/ubosstg/

Destaging area path: /home/viadmin2/ubosdata

Click **Next** to continue.

Figure 95. TaskGuide Object Server: Enter Setup Information

7. You can confirm the information you have supplied or click on **Back** to review the OSR3 database information on the Object Server Confirm OSR3 Database Information screen, Figure 96 on page 128. Click on **Next** to continue.



IBM Installation TaskGuide

Object Server: Confirm OSR3 Database Information

The Installation TaskGuide will install and set up your system with the following settings:

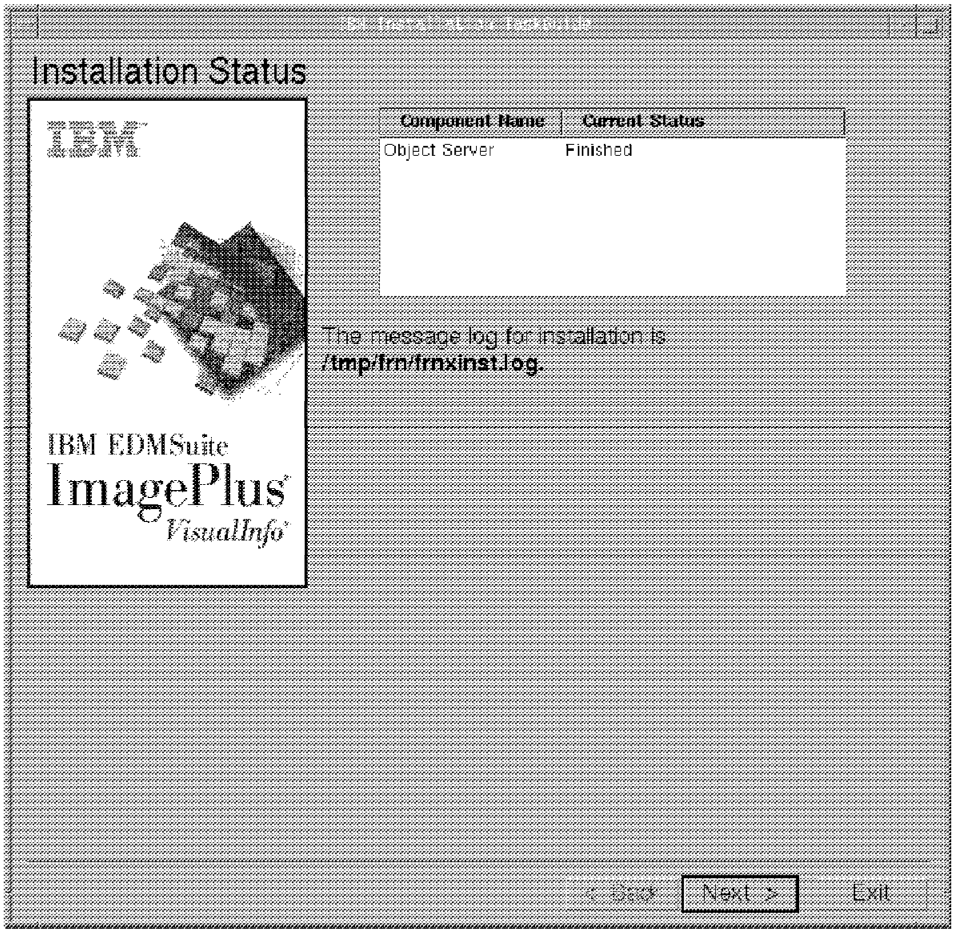
Name	Setting
DB2 Instance	vidb2in
Database Name	OSR3
Database Description	Object_Server_Database
Associated Library Server	LSR3
Staging Area Path	/home/viadmin2/ubosstg/
Destaging Area Path	/home/viadmin2/ubosdata/

To change the settings, click **Back** to return to the previous panels and then type over the value that you want to change.

Click **Next** to accept the settings and install Object Server.

Figure 96. TaskGuide Object Server: Confirm OSR3 Database Information

8. The installation is successful if the Installation Status screen, Figure 97 on page 129, shows “Finished” under current status. If not successful, exit the TaskGuide and check the /tmp/frn/frnxinst.log file to see possible reasons for the failure.



Installation Status

Component Name	Current Status
Object Server	Finished

The message log for installation is **/tmp/frn/frnxinst.log.**

Figure 97. VisualInfo TaskGuide Object Server: Installation Status

9. Click on **Exit** to finish the installation.

6.4.2 Setting Up VisualInfo

We will set up the AIX components by running configuration and setup scripts, as well as defining network connections.

6.4.2.1 Setting Up a Library Server Environment

Follow the steps listed to set up the library server environment.

1. Log in as `viadmin`, the administrative user of the library server.
2. Change the directory to /usr/lpp/frn/bin.
3. To create library server setup script, enter:

   ```
   $ ./frnconfig.lib
   ```

   On the following screen, enter the database name to be used by the library server, when prompted. The database name in our example is LSR3.

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
(C) COPYRIGHT IBM CORP. 1994, 1997.
ALL RIGHTS RESERVED.
US GOVERNMENT USERS RESTRICTED RIGHTS
USE, DUPLICATION OR DISCLOSURE RESTRICTED
BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
LICENSED MATERIALS - PROPERTY OF IBM

You have not supplied a system-wide unique database name
  for the Library server component you are configuring.
You may select the default of LIBSRVRX,
  or you may enter a new database name now.

Enter the unique database name you wish to use,
  or press enter to choose the default of LIBSRVRX: LSR3

The following settings and directories are used
for this Library server component.
    Component Type:         LIBSERV
    Unique Component Id:    LSR3
    Runtime configuration:  IBMCONFIG
    Database Name:          LSR3
    Install directory:      /usr/lpp/frn
    Shared product files:   /var/frn
    Persistent files:       /viadmin/frn
To proceed, press Enter; otherwise enter any character to terminate
configuration.

- Setup script created /home/viadmin/frn/frnsetup.lib configuration for
Library server completed successfully.

OK , To set up the environment for the Library server Component run,
 ./home/viadmin/frn/frnsetup.lib
```

4. Change the directory to /home/viadmin/frn.

5. To set up the environment variables for the library server administrative user, enter:

   `$ . ./frnsetup.lib`

6. You receive a message saying that the environment is set for LSR3.

```
OK, environment set for Library server 'LSR3'  component.
```

7. To see the environment variables, enter the env command.

6.4.2.2 Setting Up an Object Server Environment

Follow these steps to set up the object server environment.

1. Log in as viadmin2, the admin user for the Object Server.

2. Change the directory to /usr/lpp/frn/bin.

3. To create the object server setup script, enter:

   `$ ./frnconfig.obj`

4. On the following screen, enter the database name to be used by the object server, when prompted. The database name in our example is OSR3.

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
 (C) COPYRIGHT IBM CORP. 1994, 1997.
 ALL RIGHTS RESERVED.
 US GOVERNMENT USERS RESTRICTED RIGHTS
 USE, DUPLICATION OR DISCLOSURE RESTRICTED
 BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
 LICENSED MATERIALS - PROPERTY OF IBM

 You have not supplied a system-wide unique database name
   for the Object server component you are configuring.
 You may select the default of OBJSRVRX,
   or you may enter a new database name now.
 Enter the unique database name you wish to use,

  Enter the unique database name you wish to use,
    or press enter to choose the default of OBJSRVRX: OSR3

  The following settings and directories are used
  for this Object server component.
      Component Type:         OBJSERV
      Unique Component Id:    OSR3
      Runtime configuration:  IBMCONFIG
      Database Name:          OSR3
      Install directory:      /usr/lpp/frn
      Shared product files:   /var/frn
      Persistent files:       /viadmin2/frn
To proceed, press Enter; otherwise enter any character to terminate
configuration.

-Setup script created /home/viadmin2/frn/frnsetup.obj for Object Server
component completed successfully.

OK , To set up the environment for the Object server Component run,
 . /home/viadmin2/frn/frnsetup.obj
```

5. Change the directory to /home/viadmin2/frn.

6. To set up the environment for the object server user, enter:

   `$ . ./frnsetup.obj`

   The following screen appears:

```
OK, environment set for Object server 'OSR3' component.
```

7. To see the environment variables, enter the env command.

6.4.2.3 Setting Up an SMS Server

Follow these steps to configure the SMS server environment for the object server.

1. Log in as `viadmin3`, the admin user for the SMS server.

2. Change the directory to /usr/lpp/frn/bin.

3. To create the SMS server for object server setup script, enter:

   `$ ./frnconfig.objsms`

4. On the following screen enter the database name to be used by the SMS server for object server, when prompted. The database name in our example is OSR3.

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
(C) COPYRIGHT IBM CORP. 1994, 1997.
ALL RIGHTS RESERVED.
US GOVERNMENT USERS RESTRICTED RIGHTS
USE, DUPLICATION OR DISCLOSURE RESTRICTED
BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
LICENSED MATERIALS - PROPERTY OF IBM

You have not supplied a system-wide unique database name
  for the Object server SMS server component you are configuring.
You may select the default of OBJSRVRX,
  or you may enter a new database name now.

Enter the unique database name you wish to use,
  or press enter to choose the default of OBJSRVRX: OSR3
  The following settings and directories are used
  for this Object server SMS server component.
      Component Type:         OBJSMSSERV
      Unique Component Id:    OSR3
      Runtime configuration:  IBMCONFIG
      Database Name:          OSR3
      Install directory:      /usr/lpp/frn
      Shared product files:   /var/frn
      Persistent files:       /home/viadmin3/frn
 To proceed, press Enter; otherwise enter any character to terminate
 configuration.

- Setup script created: /home/viadmin3/frn/frnsetup.objsms
OK, configuration for object server SMS server completed successfully.

  OK , To set up the environment for the Object server SMS server run,
   . /home/viadmin3/frn/frnsetup.objsms
```

5. Change the directory to /home/viadmin3/frn.

6. To set up the environment for the SMS server for object server user, enter:

   ```
   $ . ./frnsetup.objsms
   ```

   The following screen appears:

   ```
   OK, environment set for Object server ′OSR3′ component.
   ```

7. To see the environment variables, enter env command.

6.4.2.4 Network Table Generation

After a successful installation of all VisualInfo components in a distributed network environment, it is important to indicate how the components are associated with each other, where they are located, and which communication protocol and port number to be used, in order to run VisualInfo. This is done through the network table.

We generate the network table on ROLM3 by logging in the platform as the library server user viadmin.

Note: It can also be done with the object server user ID. In that case, ensure that frnsetup.obj in /home/viadmin2/frn has been run to create the VisualInfo environment. The rest of the procedure is the same.

Follow these steps to generate a network table on ROLM3.

- For the library server and the object server for AIX on ROLM3:

  1. Login as the library server administrative user.

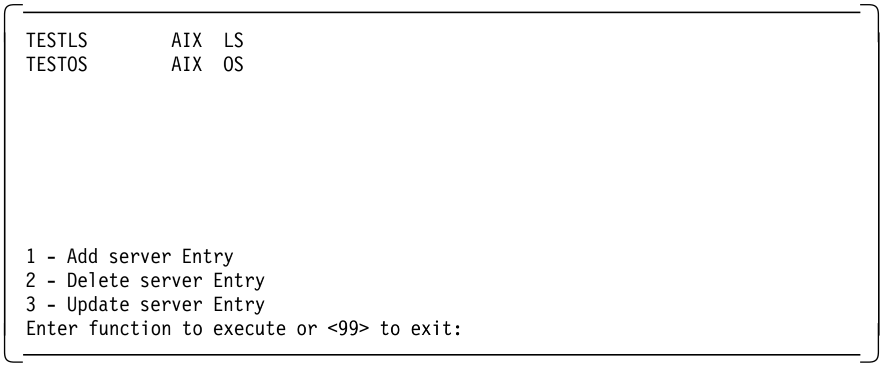
  2. Set up the environment for the library server, as follows:

```
$ . ./home/viadmin/frn/frnsetup.lib
```

3. To generate the network table, enter:

   ```
   $ frnnetcfg
   ```

   You will see the following screen:



```
TESTLS       AIX  LS
TESTOS       AIX  OS








1 - Add server Entry
2 - Delete server Entry
3 - Update server Entry
Enter function to execute or <99> to exit:
```

- From the menu:

  1. Select option **1** to add a library server entry.
  2. Select Server Location: select **1** for local server.

     Decide on the location of the server you add. It can be either local (if it is on the same platform from where you are generating the network table), or remote (if it is on some other platform and you want to add it in this network table).

  3. Select Type of Server: select **1** for library server.
  4. Select Platform Type: select **2** for AIX.
  5. Enter the name of the server: **LSR3**
  6. Enter the port number: **30000**

     This sets the port number for the library server.

  7. Select the following Options: select **1** to save the data and create an entry in the network table.
  8. Select **1** to add object server entry.
  9. Follow the rest of the steps as you followed for the library server.
  10. The port number for the object server is **30002**.

      Note: When you specify a port number for servers, the numerical difference between the port number of the library server and the object server must be a minimum of 2 (two). This is because the next port (for example 30001) is assigned for the utility services (utility server) for the library server; the object server is a similar case.

  11. This process creates a network table, file FRNOLINT.TBL, in the /var/frn directory. You can open the network table file in this directory to verify that the information you supplied is correct.

You would also generate a network table on the VisualInfo system administration client workstation to add the object server and library server. Follow these steps to generate a network table on a Windows NT client workstation (WOLM100):

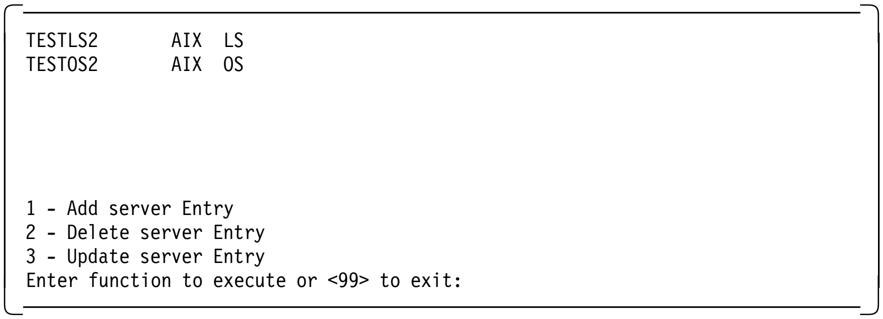
Note: In our setup, both system administration client and VisualInfo client are on the same workstation, so we created a network table once for both clients.

1. On the workstation where system administration is installed, go to the drive where it is installed and change the directory to frnroot.

2. To generate the network table, enter:

   ```
   c:\frnroot>frnnlinc
   ```

   You will see the following screen.

   ```
   TESTLS2       AIX  LS
   TESTOS2       AIX  OS







   1 - Add server Entry
   2 - Delete server Entry
   3 - Update server Entry
   Enter function to execute or <99> to exit:
   ```

3. Select option **1** to add a server entry.

4. From the Select Server Location prompt, enter **2** for remote server.

5. From the Select type of server prompt, enter **1** for library server.

6. From the Select Platform type, enter **2** for AIX.

7. From the Enter the server name prompt, enter **LSR3**.

8. From the Enter the port number prompt, enter the port number of the library server you set on the AIX library server system.

9. From the Enter the Hostname prompt, enter the hostname **ROLM3**, on which the library server is installed.

10. From the Select from the following options prompt, select **1** to add an entry for the object server entry in the network table.

11. Follow the preceding steps for the object server also. Supply the correct data for the object server when prompted.

12. This process creates a network table file, FRNOLINT.TBL, in the c:\frnroot directory.

13. For a system administrator client, you must add not only a library server but also an object server to be associated with each library server. Otherwise the network table for a system administrator is incomplete. For a client, you add only library server information to the network table.

14. Now stop the client messenger. From the VisualInfo system administration workstation, select:

    ```
    Start
     ---> Programs
           ---> VisualInfo for Windows
                  ---> Utility
                         --->Stop client Messenger
    ```

    This process ensures that the updated network table is available from now on for the application.

6.4.2.5 Network Connection Verification

After generating a network table, we have to verify that it is correct. For this purpose, we run following tests:

1. Ping from the VisualInfo system administration the library server and verify that TCP/IP communication is up.
2. Start the library server and the object server on ROLM3
3. Use the network connection verification utility provided with VisualInfo system administration.

   From the VisualInfo system administration workstation, select:

   ```
   Start
    ---> Programs
          ---> VisualInfo
                 ---> Utility
                         --->Network Connection Verification
   ```

   Wait for the message `Network Connection Verification is completed successfully` to be displayed.

6.4.2.6 Installation Verification

Follow these steps to verify that the VisualInfo installation is correct.

1. Start the library server and the object server (see 6.4.3, "Hints on Starting and Stopping VisualInfo Servers").
2. From a client workstation, log onto library server LSR3, using `frnadmin` as the client, and `password` as the password.
3. Click on **File** from the menu table.
4. Click on **File and Import** to import an object into VisualInfo. If you can import, your installation and network configuration are correct. If you cannot import, check that the C Set ++ compiler is available and licensed. See 6.3.2, "Installation of C Set ++" on page 88.

   Also, check on the availability and licensing of DB2 by creating a DB2 database. If you cannot create a database from DB2, be sure that you get the license.
5. Stop the object server and bring up the SMS server from the workstation where the object server is installed.
6. From the VisualInfo system administration workstation, log in as `frnadmin` with `password`. Try to define a storage class, an index class, and a volume. If you can define these, the VisualInfo setup is correct.
7. If you have problems while completing these steps, open the log file frndiag.log under the /home/viadmin/frn/log directory, and check for reasons why the error occurred.

6.4.3 Hints on Starting and Stopping VisualInfo Servers

In this section we provide steps to start and stop the object server, the library server and the SMS server. The object server must be shut down properly. If the object server has been stopped improperly, you cannot restart it without recovering the database. There is a utility for this; the icon is in the system administration utilities folder. The time required and result depends primarily on the size of the database.

Follow these steps for starting and stopping the servers properly.

6.4.3.1 Starting the Object Server

1. Log in as `viadmin2`, the administrative user of the object server.

2. Change the directory to /home/viadmin2/frn.

3. To set up the environment for the object server administrative user, enter:

   ```
   $ . ./frnsetup.obj
   ```

4. Start the object server in the same shell and session where the object server environment is set up.

   ```
   $ frnstart.obj
   ```

   Recommendation: To get your prompt back, include an ampersand (&) in the command. To enable you to exit the osadmin ID and not kill your object server process, include the parameter nohup. The command would look like this:

   ```
   # nohup frnstart.obj &
   ```

   The output should be:

   ```
   VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
   (C) COPYRIGHT IBM CORP. 1994, 1997.
   ALL RIGHTS RESERVED.
   US GOVERNMENT USERS RESTRICTED RIGHTS
   USE, DUPLICATION OR DISCLOSURE RESTRICTED
   BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
   LICENSED MATERIALS - PROPERTY OF IBM


   Starting Object server with database ’OSR3’ and configuration ’IBMCONFIG
   FRN2310I: (OSR3) server status : Initializing
   FRN2311I: (OSR3) server status : Accepting requests
   ```

5. Check for the server status `Accepting requests`.

6.4.3.2 Stopping the Object Server

1. Press ctrl+z in the object server session to get the command prompt.

2. Put the server process in the background by entering:

   ```
   $ bg
   ```

   ```
   (1)     frnstart.obj&
   ```

3. Enter the command:

   ```
   $ frncmd
   ```

4. You will see the following screen:

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
(C) COPYRIGHT IBM CORP. 1994, 1997.
ALL RIGHTS RESERVED.
US GOVERNMENT USERS RESTRICTED RIGHTS
USE, DUPLICATION OR DISCLOSURE RESTRICTED
BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
LICENSED MATERIALS - PROPERTY OF IBM


Enter a command:
        Help
        Connect {server name}
 Listservers
         Quit
 >>
```

5. Connect to the object server (OSR3):

   `>>connect OSR3`

6. Enter the `help` command at the > > prompt to receive the following menu:

```
Using Server: osr3
Enter a command:
        Help
        Status
        Migrator {Status|START|Enable|Disable}
        Destager {Status|START|Enable|Disable}
        Purger   {Status|START|Enable|Disable}
        Trace {Status|ON|OFF}
        SHutdown
        DIsconnect
        Quit
```

7. Enter the `shutdown` command to bring the server down. Enter **1** to confirm, and wait for the message indicating `Server status : Down.`

```
>> shutdown

FRN2032I: Enter 1 to confirm that you want to shutdown, enter
          anything other than 1 to cancel shutdown.
FRN2313I: (OSR3) server status : Closing

FRN2033I: server shutdown initiated.

>> FRN2314I: (OSR3) server status : Down
```

8. The object server is now properly shut down.

9. Optionally, if there are orphaned processes left running, run the following command: `$ frnkill.all`

10. We recommend using this procedure for proper shutdown, so keep the session accessible at the workstation.

If you need to clean up the object server processes, here is another procedure to use.

1. As root user, issue:

   `ps-ef | grep frn`

   If you see any active processes owned by viadmin1 (or 2 or 3), then VisualInfo was not stopped properly with the frncmd utility.

2. These resources can be cleaned up by issuing the command from the viadmin2 user ID:

```
$ frnkill.all
```

The output should be:

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
(C) COPYRIGHT IBM CORP. 1994, 1997.
ALL RIGHTS RESERVED.
US GOVERNMENT USERS RESTRICTED RIGHTS
USE, DUPLICATION OR DISCLOSURE RESTRICTED
BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
LICENSED MATERIALS - PROPERTY OF IBM


 Cleaning up system resources.  Please wait.
 Stopping processes.  Please wait.
 Cleaning up system resources.  Please wait.
 Successfully stopped.
```

3. You must then run the object server recovery program.

6.4.3.3 Starting the Library Server

1. Log in as `viadmin`, the administrative user for the library server.

2. Change the directory to /home/viadmin/frn.

3. Run the following script to set up the environment for library server administrative user.

   ```
   $ . ./frnsetup.lib
   ```

4. To start the library server, enter:

   ```
   $ frnstart.lib
   ```

   The output should be:

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
(C) COPYRIGHT IBM CORP. 1994, 1997.
ALL RIGHTS RESERVED.
US GOVERNMENT USERS RESTRICTED RIGHTS
USE, DUPLICATION OR DISCLOSURE RESTRICTED
BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
LICENSED MATERIALS - PROPERTY OF IBM


Starting Library server with database 'LSR3' and configuration 'IBMCONFIG
FRN2310I: (LSR3) server status : Initializing
FRN2311I: (LSR3) server status : Accepting requests
```

5. Check for the server status message `Accepting requests`.

6. To see the library server status, enter the following command from the library server session: `frncmd`

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
(C) COPYRIGHT IBM CORP. 1994, 1997.
ALL RIGHTS RESERVED.
US GOVERNMENT USERS RESTRICTED RIGHTS
USE, DUPLICATION OR DISCLOSURE RESTRICTED
BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
LICENSED MATERIALS - PROPERTY OF IBM


Enter a command:
        Help
        Connect {server name}
        Listservers
        Quit
>>
```

7. Connect to the library server as: `>> connect LSR3`
8. Enter the `help` command at the > > prompt to get this menu:

```
Using Server: lsr3
Enter a command:
        Help
        Status
        Trace {Status|ON|OFF}
        SHutdown
        DIsconnect
        Quit
```

You can now use the interactive menu.

6.4.3.4 Stopping the Library Server

1. To stop the library server now (in the same session), enter: `>> shutdown`
2. Enter 1 to confirm the shutdown:

```
>> shutdown

        FRN2032I: Enter 1 to confirm that you want to shutdown, enter anything
other than 1 to cancel shutdown.
FRN2313I: (LSR3) server status : Closing

        FRN2033I: server shutdown initiated.

>> FRN2314I: (LSR3) server status : Down
```

3. The library server is properly shut down.
4. We recommend using this procedure for proper shutdown, so keep the session accessible at the workstation.

   If you need to clean up the library server processes, here is another procedure to use.

5. As root user, issue:

   ```
   ps-ef | grep frn
   ```

   Any active processes owned by viadmin1 (or 2 or 3) indicate that VisualInfo was not stopped properly with the frncmd utility.

6. These resources can be cleaned up by issuing the command from the viadmin2 user ID:

   ```
   $ frnkill.all
   ```

   The output should be:

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
(C) COPYRIGHT IBM CORP. 1994, 1997.
ALL RIGHTS RESERVED.
US GOVERNMENT USERS RESTRICTED RIGHTS
USE, DUPLICATION OR DISCLOSURE RESTRICTED
BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
LICENSED MATERIALS - PROPERTY OF IBM


 Cleaning up system resources.  Please wait.
 Stopping processes.  Please wait.
 Cleaning up system resources.  Please wait.
 Successfully stopped.
```

7. Cleanup system resources by running:

   `$ frnkill.all`

   The output should be:

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
(C) COPYRIGHT IBM CORP. 1994, 1997.
ALL RIGHTS RESERVED.
US GOVERNMENT USERS RESTRICTED RIGHTS
USE, DUPLICATION OR DISCLOSURE RESTRICTED
BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
LICENSED MATERIALS - PROPERTY OF IBM


 Cleaning up system resources.  Please wait.
 Stopping processes.  Please wait.
 Cleaning up system resources.  Please wait.
 Successfully stopped.
```

6.4.3.5 Starting the SMS Server for Object Server

1. Log on as the user of the SMS Server, `viadmin3`.
2. Change the directory to /home/viadmin3/frn.
3. Run the following script to set up the environment for the SMS server user.

   `$ frnsetup.objsms`

```
OK, environment set for the Object server SMS server ’OSR3’ component.
```

4. Start the SMS Server: `$ frnstart.objsms`

```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
 (C) COPYRIGHT IBM CORP. 1994, 1997.
 ALL RIGHTS RESERVED.
 US GOVERNMENT USERS RESTRICTED RIGHTS
 USE, DUPLICATION OR DISCLOSURE RESTRICTED
 BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
 LICENSED MATERIALS - PROPERTY OF IBM


FRN2057I: The SMS server is active.

FRN2055I: If you want to stop the SMS server, type stop and press Enter.
```

5. Check for a server status of `active`.

6.4.3.6 Stopping the SMS Server for Object Server

1. If you want to stop the SMS server, enter: `>>stop` from the SMS server window.

If you need to clean up the SMS server processes, here is another procedure to use.

1. As root user, issue:

   ```
   ps-ef | grep frn
   ```

   Any active processes owned by viadmin1 (or 2 or 3) indicate that VisualInfo was not stopped properly with the frncmd utility.

2. Set up the environment for the SMS server for object server:

   ```
   $ frnsetup.objsms
   ```

3. Enter `$ frnkill.all`

4. A message will confirm the server is stopped.

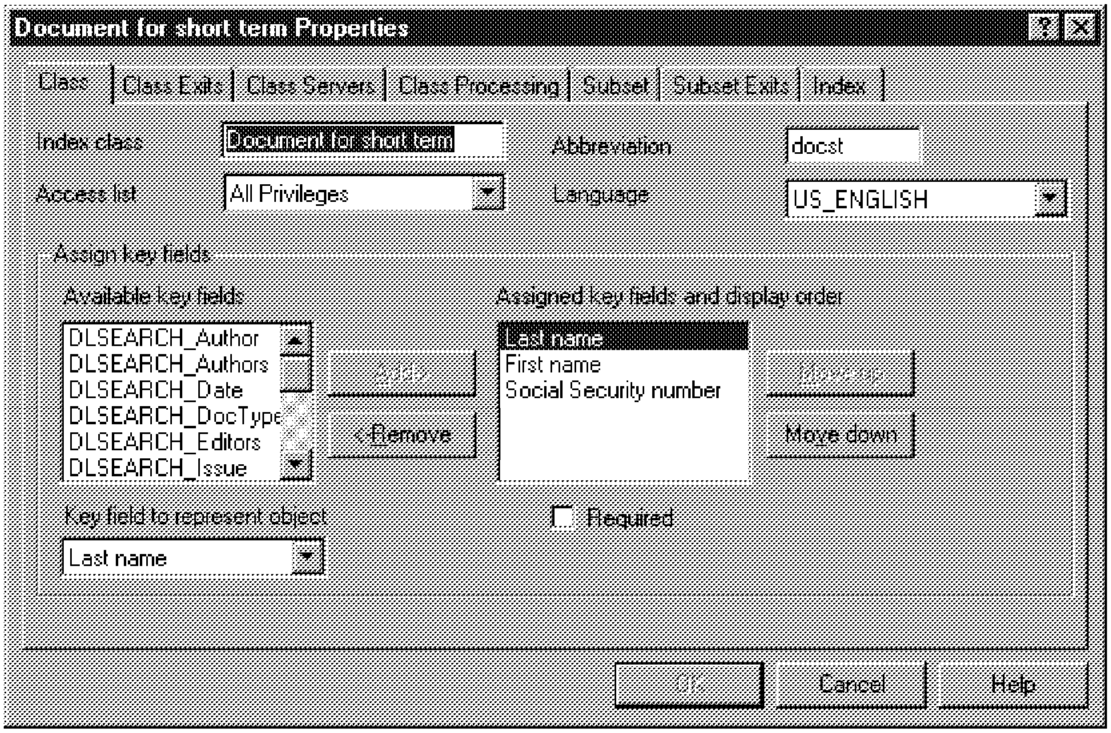
```
VISUALINFO VERSION 2 RELEASE 3.0 - 5697-250.
(C) COPYRIGHT IBM CORP. 1994, 1997.
ALL RIGHTS RESERVED.
US GOVERNMENT USERS RESTRICTED RIGHTS
USE, DUPLICATION OR DISCLOSURE RESTRICTED
BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
LICENSED MATERIALS - PROPERTY OF IBM


 Cleaning up system resources.  Please wait.
 Stopping processes.  Please wait.
 Cleaning up system resources.  Please wait.
 Successfully stopped.
```

6.4.4 Configuring VisualInfo

Most VisualInfo resources required for this SMS configuration (Example 2) are set up by default during the VisualInfo installation. The only resource we have to define is the *index* class, where we specify the name of the collection (SMS resource) to use during the indexing operation. We create the following two index classes and associate them with their related collection names.

- Index Class: Document for short-term --> Collection: CNST
- Index Class: Document for long-term --> Collection: CNLT

Figure 98 on page 142 and Figure 99 on page 142 show the configuration panels we used to configure the document for a short-term index class.



Index class: Document for short term

Abbreviation: docst

Access list: All Privileges

Language: US_ENGLISH

Assign key fields

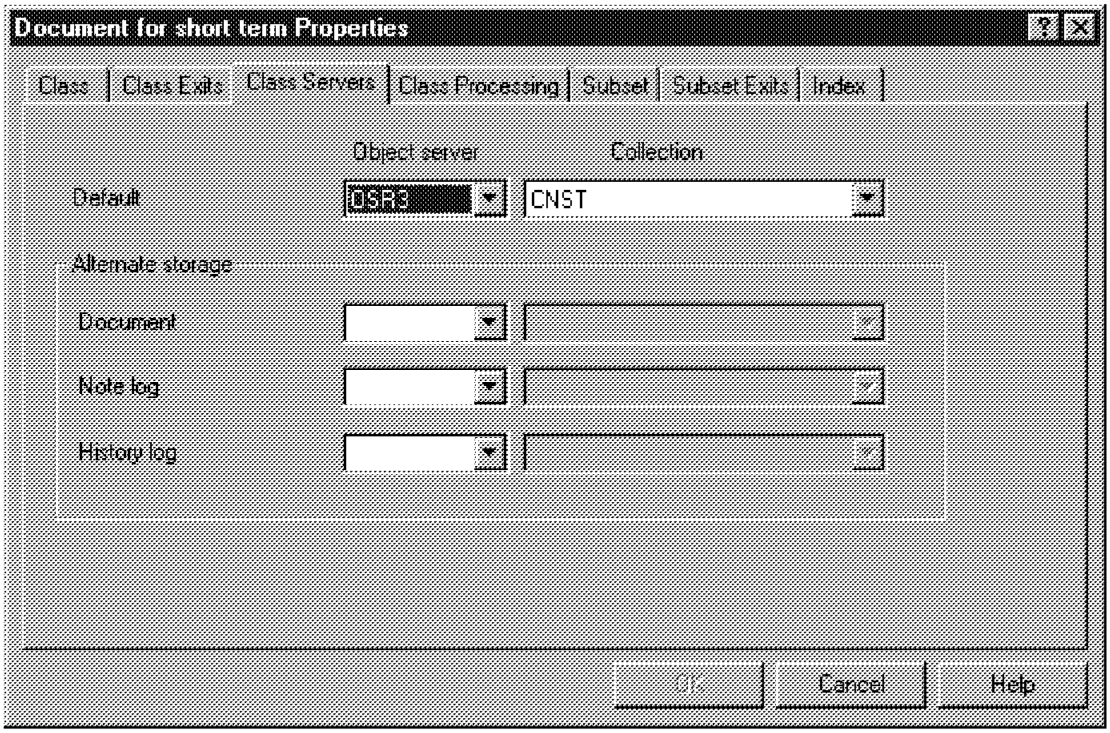
Available key fields: DLSEARCH_Author, DLSEARCH_Authors, DLSEARCH_Date, DLSEARCH_DocType, DLSEARCH_Editors, DLSEARCH_Issue

Assigned key fields and display order: Last name, First name, Social Security number

Key field to represent object: Last name

□ Required

Figure 98. Document for Class Configuration of Short-Term Index Class



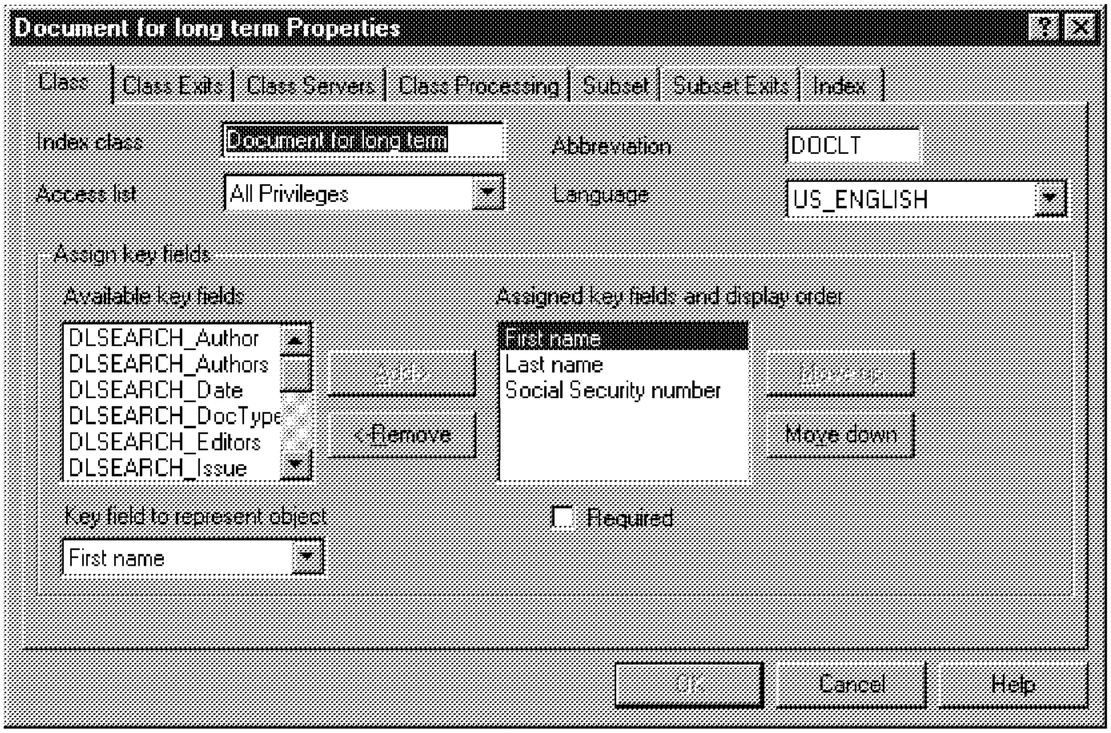
	Object server	Collection
Default	OSR3	CNST

Alternate storage

Document		
Note log		
History log		

Figure 99. Document for Class Servers Configuration of Short-Term Index Class

Figure 100 on page 143 and Figure 101 on page 143 show the configuration panels we used to configure the document for a long-term index class.



Index class: Document for long term

Abbreviation: DOCLT

Access list: All Privileges

Language: US_ENGLISH

Assign key fields

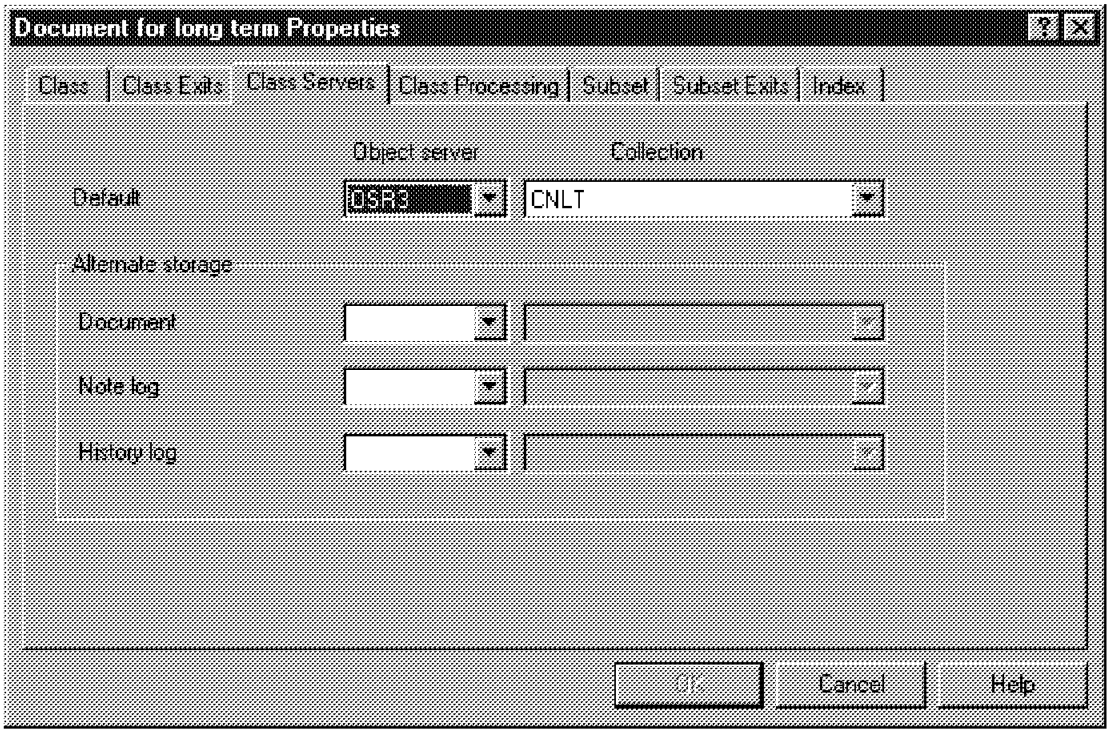
Available key fields:
DLSEARCH_Author
DLSEARCH_Authors
DLSEARCH_Date
DLSEARCH_DocType
DLSEARCH_Editors
DLSEARCH_Issue

Assigned key fields and display order:
First name
Last name
Social Security number

Key field to represent object: First name

□ Required

Figure 100. Document for Class Configuration of Long-Term Index Class



	Object server	Collection
Default	OSR3	CNLT

Alternate storage

Document		
Note log		
History log		

Figure 101. Document for Class Servers Configuration of Long-Term Index Class

6.4.5 Configuring SMS

In this section we present the steps to configure the SMS resources on ROLM3 for Example 2.

The SMS resource definitions are critical. They must be created in the following order:

1. Device manager
2. Storage class
3. Management class
4. Volume
5. Storage group
6. Collection
7. Staging area
8. Configuration

Note: We recommend that you not delete any default SMS resources. If your system enters an error condition, these default resources can help you recover.

The SMS configuration operations are executed on the VisualInfo system administration platform WOLM100. When VisualInfo is installed, default SMS resources are created automatically. Of the default SMS resources, we used only the device manager. We created all of the other resources listed in Table 5. (See 4.2.4, “ADSM, VisualInfo Fileroom and SMS Component Definitions” on page 59 for their definition.)

Table 5. SMS Resources to Define

SMS Resources	**Name**
Device manager	FRNXADDM, FRNXHDDM
Storage class	SCDAST, SCDALT, SCOPST, SCOPLT
Management class	MCVIST, MCVILT
Volume	/SHORTTERM, /LONGTERM, ADSM: MCE2ST, ADSM: MCE2LT
Storage group	SGST, SGLT
Collection	CNST, CNLT
Staging area configuration	
Object server configuration	EX2CFG

Figure 102 on page 145 shows the relationships between the SMS resources for collection CNST. The relationships are the same for collection CNLT.

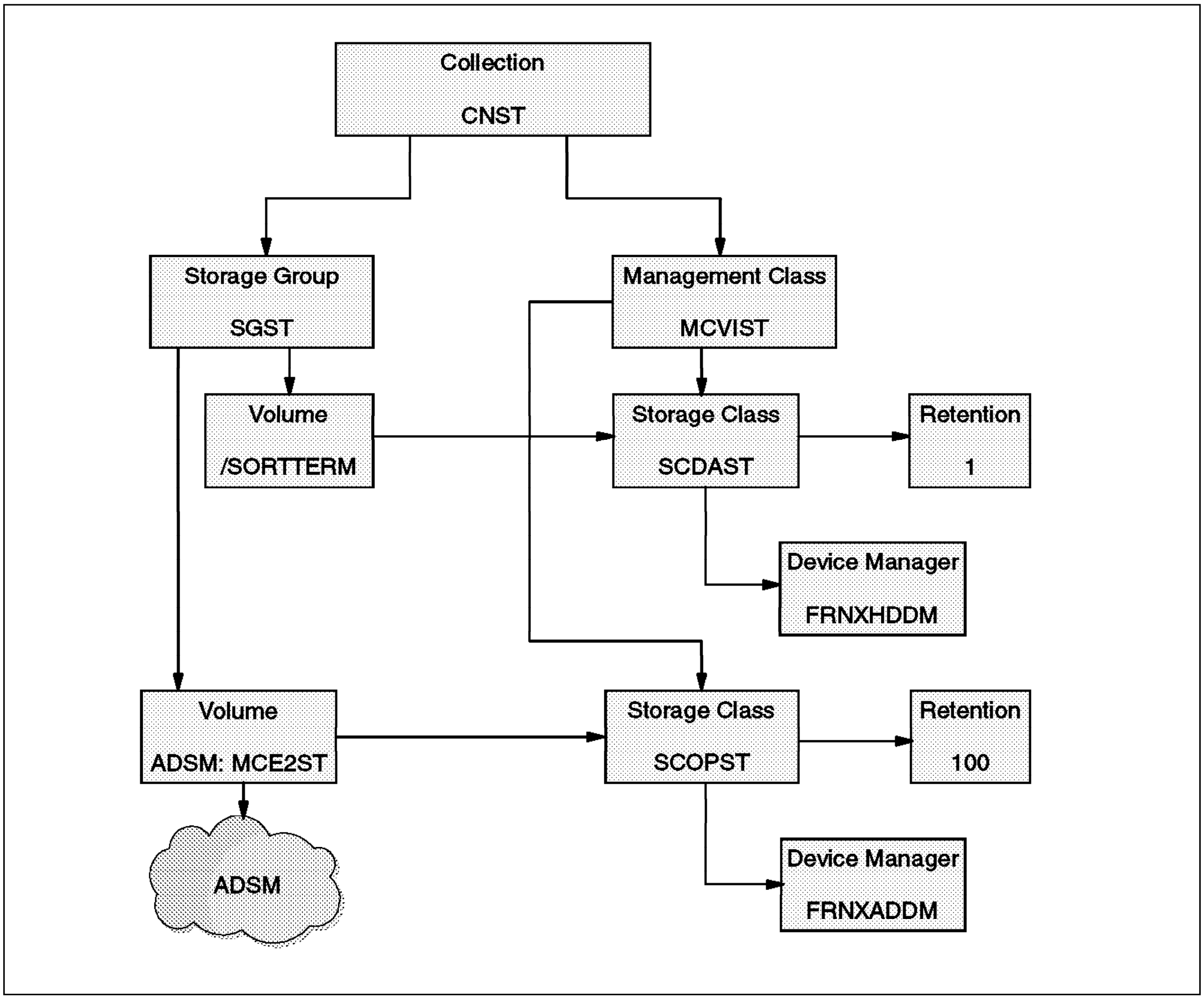


Figure 102. SMS Resources Relationship

6.4.5.1 Prerequisite Steps on ROLM3

In our Example 2, documents are first stored on DASD and then migrated to the optical library. For the first storage operation, SMS uses file systems to transfer in documents it receives from the staging area.

These file systems must be accessed in read/write mode by the viadmin2 user ID, which is the object server database administrator, so we have to change the owner of the file systems once they are created by the AIX administrator. For our configuration, we had to define the following file systems:

- /SHORTTERM (for storing documents with a short life span)
- /LONGTERM (for storing documents with a long life span)

We could have registered both types of documents in the same file system, because the lifetime has meaning only for the last storage location, but we chose to split them as they are indexed.

To create the /SHORTTERM file system, mount it, and allow the viadmin2 user ID to use it, follow this procedure:

1. Log on to ROLM3 with the root user ID and execute the following command:

   ```
   #smit jfs
   ```

2. When the System Management Interface Tool : root@rolm3 window appears, select the following options:

   ```
   Add a Journaled File System
       --> Add a Standard Journaled File System
             --> rootvg
   ```

   (rootvg is an example of the volume group to use to store this file system.)

3. When the Add a Standard Journaled File System : root@rolm3 window appears, complete the following fields as indicated:

   - **Size of file system (in 512 byte blocks)** --> 1000000 (this size should fit the need to store documents that are handled every day.)
   - **MOUNT POINT** --> /SHORTTERM
   - **Mount AUTOMATICALLY at system restart?** --> Yes
   - **PERMISSIONS** --> read/write

4. Click on **OK**, and wait until the command completes successfully.

5. To ensure that the object server database administrator can write inside this file system, execute the following command:

   ```
   #chown viadmin2 /SHORTTERM
   ```

Use the same procedure to create the /LONGTERM file system.

6.4.5.2 Configuration Steps

Important

Once VisualInfo has been installed, the first step is to create a new VisualInfo user with the MAXPRIVS privileges as a backup, just in case problems occur with the default VisualInfo administrator, FRNADMIN.

Before configuring SMS from the VisualInfo system administration workstation, you must do the following:

- Start the SMS server on the ROLM3 AIX server.
- Stop the object server on ROLM3.

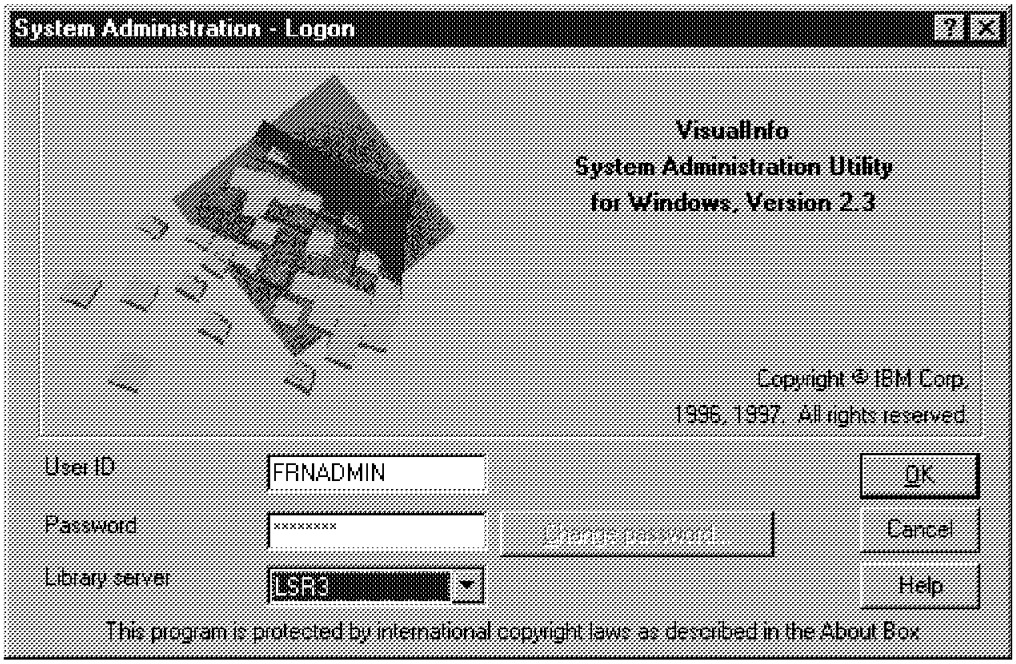
Otherwise severe errors could occur during the registration of the new SMS components in the object server database.

To log on to the System Administration platform, run the following procedure:

1. From the System Administration platform WOLM100, execute the VisualInfo System Administration program by selecting the following from the main Windows NT desktop:

   ```
   Start
        --> Program
                 -->  VisualInfo Windows
                                   --> System Administration
   ```

2. The System Administration-Logon window appears. Connect to the Library server LSR3 on ROLM3 with the administrator user ID provided (FRNADMIN) and its password (password). See Figure 103 on page 147.



System Administration - Logon

VisualInfo
System Administration Utility
for Windows, Version 2.3

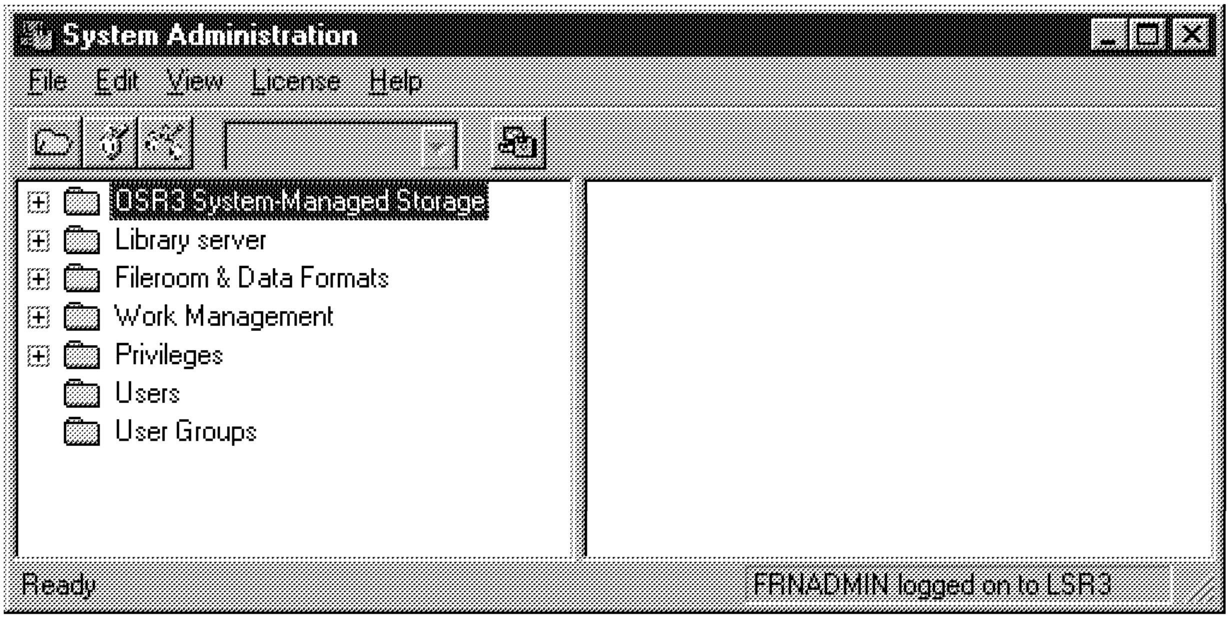
Copyright © IBM Corp.
1996, 1997. All rights reserved.

User ID: FRNADMIN
Password: ********
Library server: LSR3

This program is protected by international copyright laws as described in the About Box

Figure 103. System Administration - Logon

3. Click on **OK.** The System Administration container window menu is displayed (see Figure 104).



System Administration

- OSR3 System-Managed Storage
- Library server
- Fileroom & Data Formats
- Work Management
- Privileges
- Users
- User Groups

Ready

FRNADMIN logged on to LSR3

Figure 104. System Administration Container Window

The System Administration window contains two panes. The left pane lists the folders containing the different system administration functions. To work with a function, we double-click on its folder.

The right pane list the objects in the selected folder.

Double-click on the OSR3 System Managed Storage folder to start the configuration of object server OSR3 components on ROLM3. The related subfolders are then displayed (see Figure 105 on page 148). Define each of them in turn.

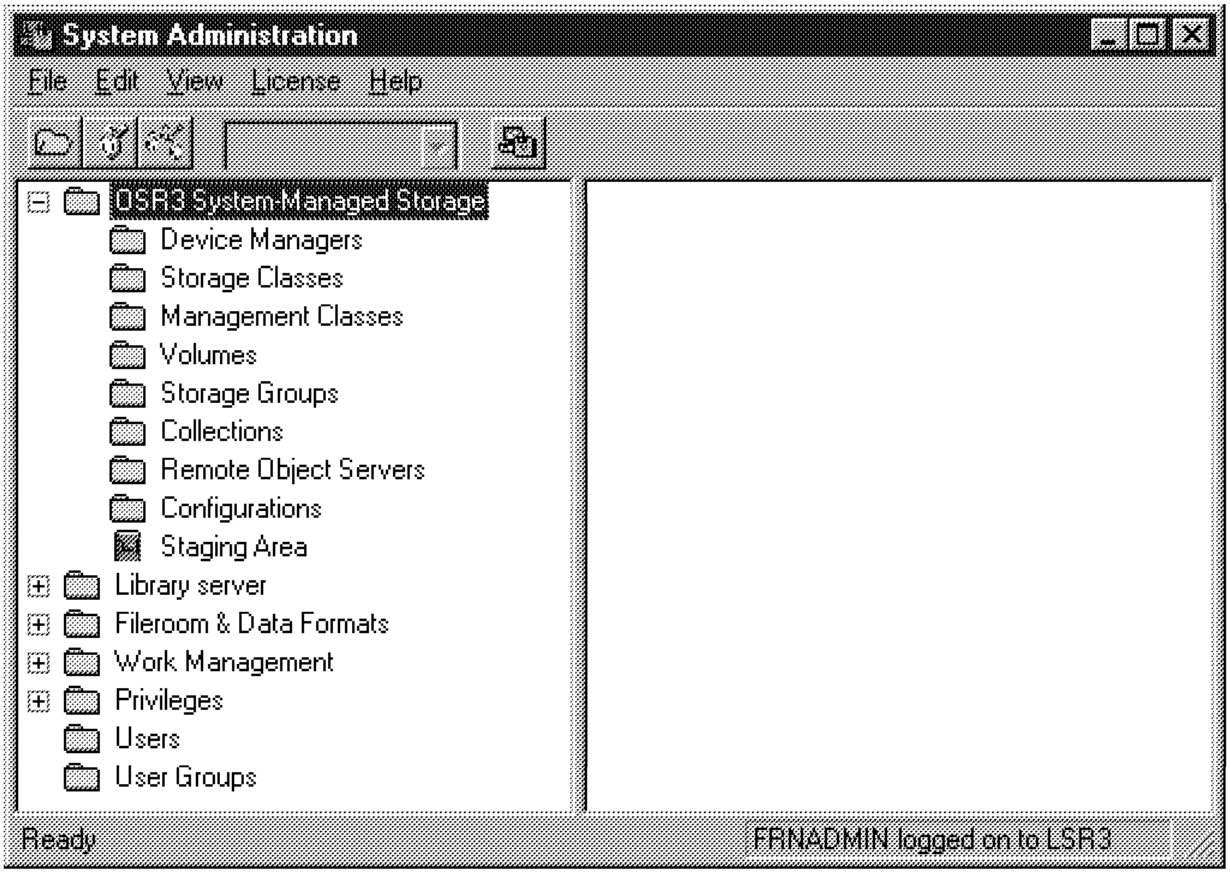


Figure 105. SMS Components List

1. Device Manager

   The device manager is the interface between the device drivers and the applications that need to access the devices.

   Double-click on the Device Managers subfolder. The three device managers installed by default are listed in the right-hand pane. See Figure 106 on page 149.

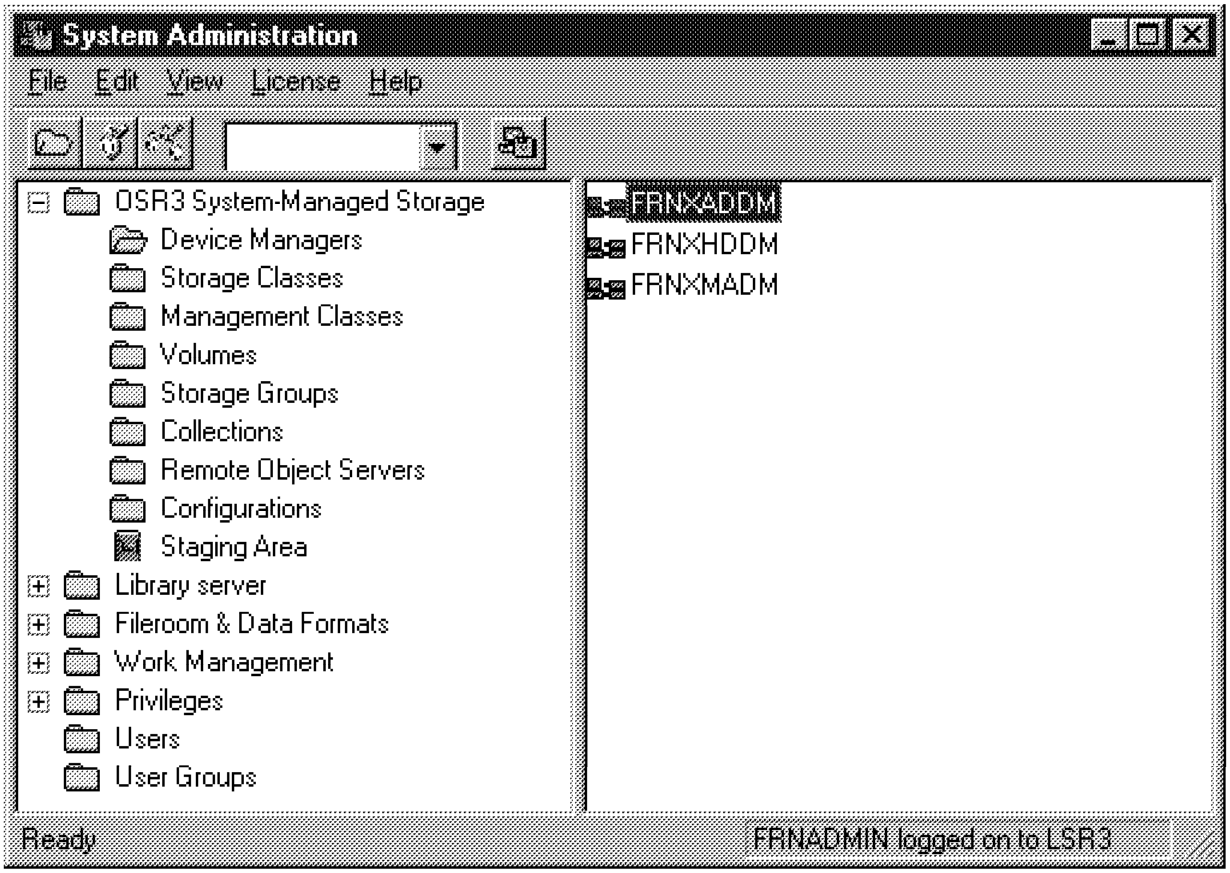
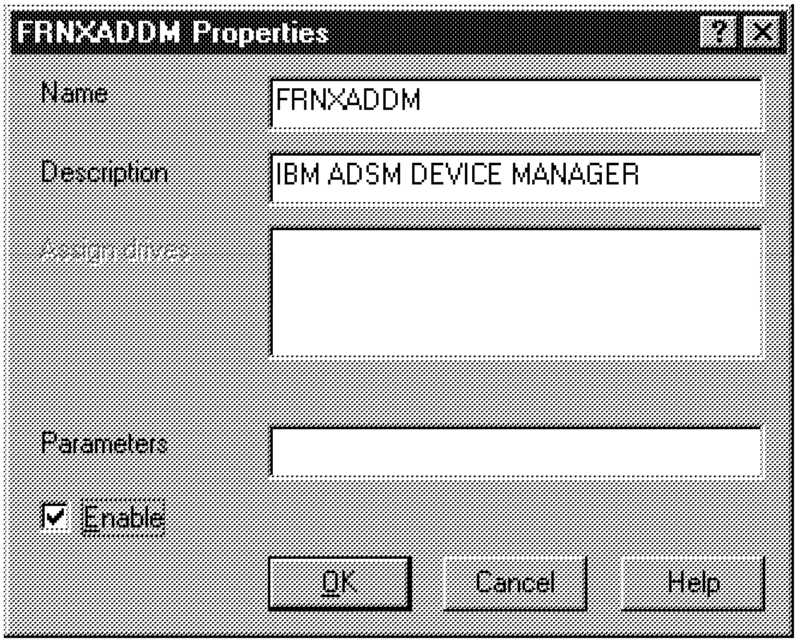


Figure 106. Device Manager List

- FRNXADDM is the device manager for ADSM.
- FRNXHDDM is the device manager for fixed disk.
- FRNXMADM is the device manager for tape.

For our configuration, only the two first Device Managers are used. By default FRNXADDM is installed but disabled. To allow VisualInfo to interface with ADSM, we must enable it. We double-click on its related object, and check the Enable box as shown in Figure 107 on page 150. Then click on **OK**.

Note: Run this step only if the ADSM client is installed on the VisualInfo server.



FRNXADDM Properties

Name: FRNXADDM

Description: IBM ADSM DEVICE MANAGER

Assign drives:

Parameters:

☑ Enable

Figure 107. ADSM Device Manager

2. Storage Class

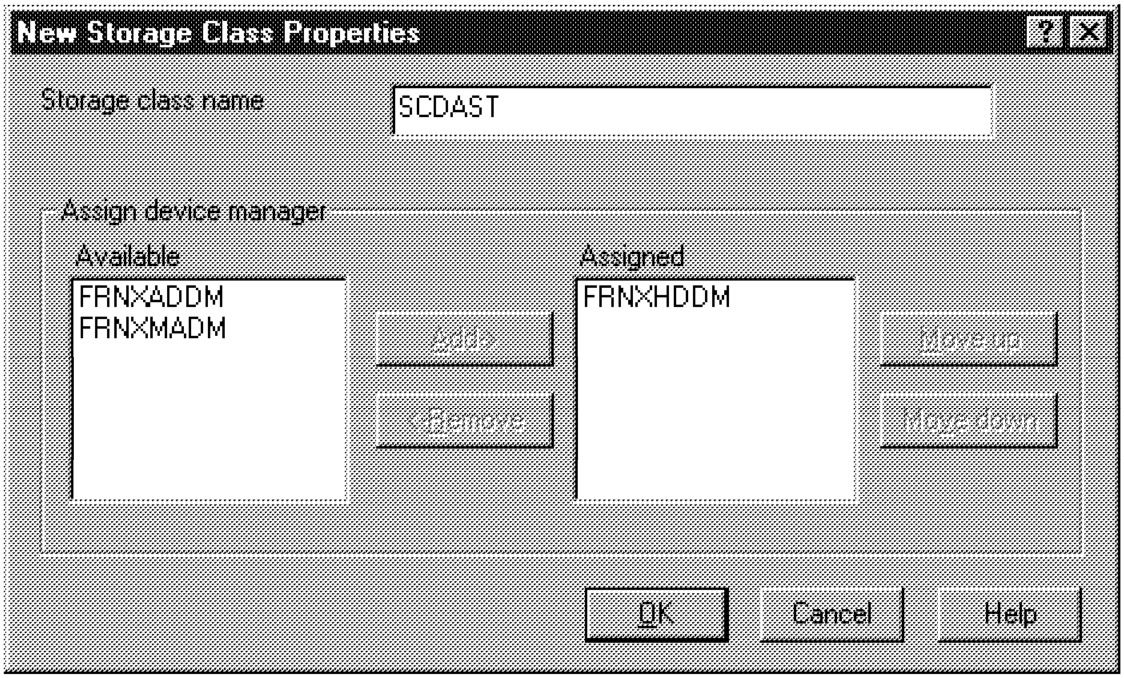
   A *storage class* consists of a storage media device manager that stores objects on volumes of its media type.

   Return to the SMS components list (Figure 105 on page 148) and double-click on the Storage Classes folder. The storage classes installed by default are listed in the right-hand pane. None of them fit our requirements, so we have to create our own storage classes.

   To create the first storage class, SCDAST, click with the right-hand mouse button with the Storage Class folder selected, and select:

   ```
   --> New
      --> Storage Class
   ```

   The New Storage Class Properties window appears (see Figure 108 on page 151). Select its appropriate device manager (FRNXADDM) in the available box, add it to the Assigned box, and register its name.



Storage class name: SCDAST

Available: FRNXADDM, FRNXMADM

Assigned: FRNXHDDM

Figure 108. Storage Class Creation

Click on **OK** to register this storage class.

Repeat this step to create the three other storage classes, as specified in our configuration, with their appropriate device manager:

SCDALT --> FRNHDDM SCOPST --> FRNADDM SCOPLT --> FRNADDM

3. Management Class

A *management class* is a logical entity that defines the policy for migrating objects.

Return to the SMS components list (Figure 105 on page 148) and double-click on the Management Classes folder. The default management classes are listed in the right-hand pane. None of them fit our requirements, so we have to create our own management classes.

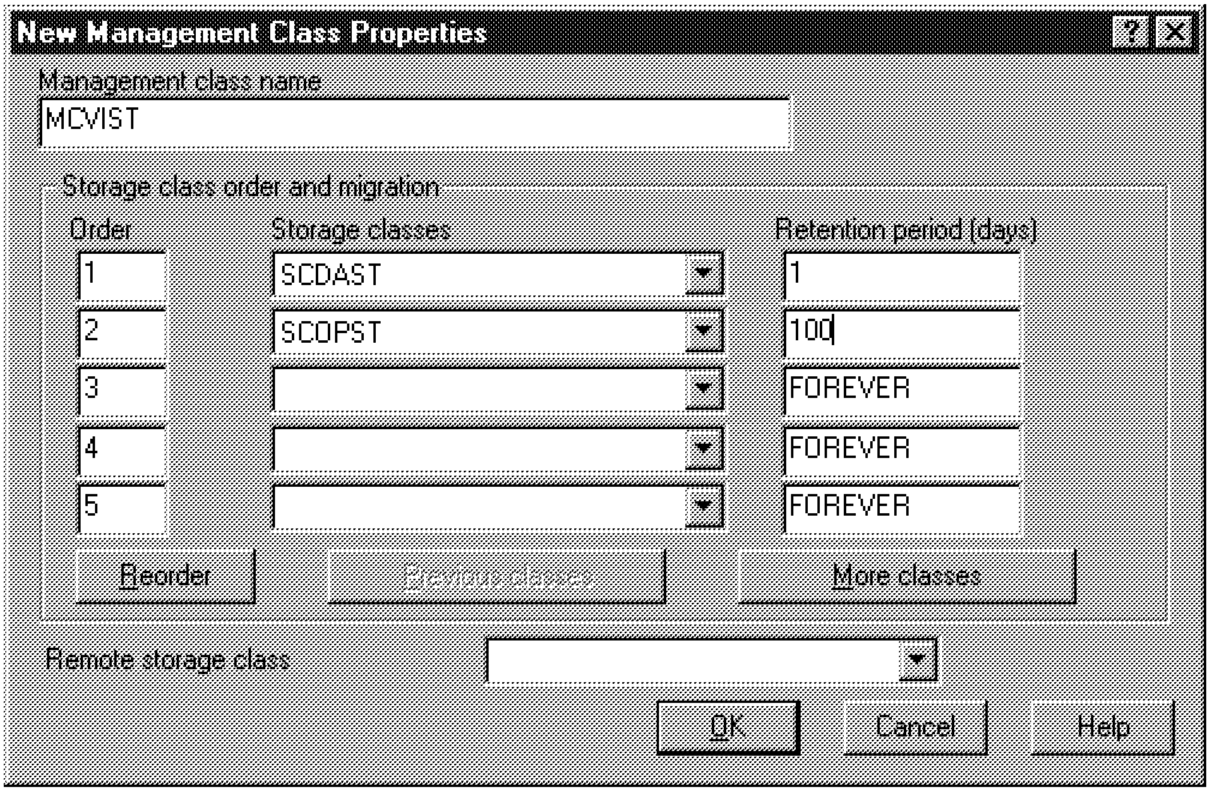
To create the first management class, MCVIST, click on the right mouse button with the Management Classes folder selected and select:

--> New

The New Management Class Properties window appears. Select its appropriate storage class

SCDAST, order 1 for 1 day
SCOPST, order 2 for 100 days

and register its name (see Figure 109 on page 152).



Management class name: MCVIST

Storage class order and migration

Order	Storage classes	Retention period (days)
1	SCDAST	1
2	SCOPST	100
3		FOREVER
4		FOREVER
5		FOREVER

Remote storage class:

Figure 109. Management Class Creation

Click on **OK** to register this management class.

Repeat this step to create the other management classes, as specified in our configuration, with their appropriate storage classes:

```
MCVILT  --> SCDALT, order 1 for 1 day
        --> SCOPLT, order 2 for 100 days
```

For both management classes, select a retention period of one day on DASD to verify that the migration actually occurs the day following the indexing process.

4. Volume

A *volume* is a physical storage medium that a storage class manages.

The name of the volume is defined directly by SMS:

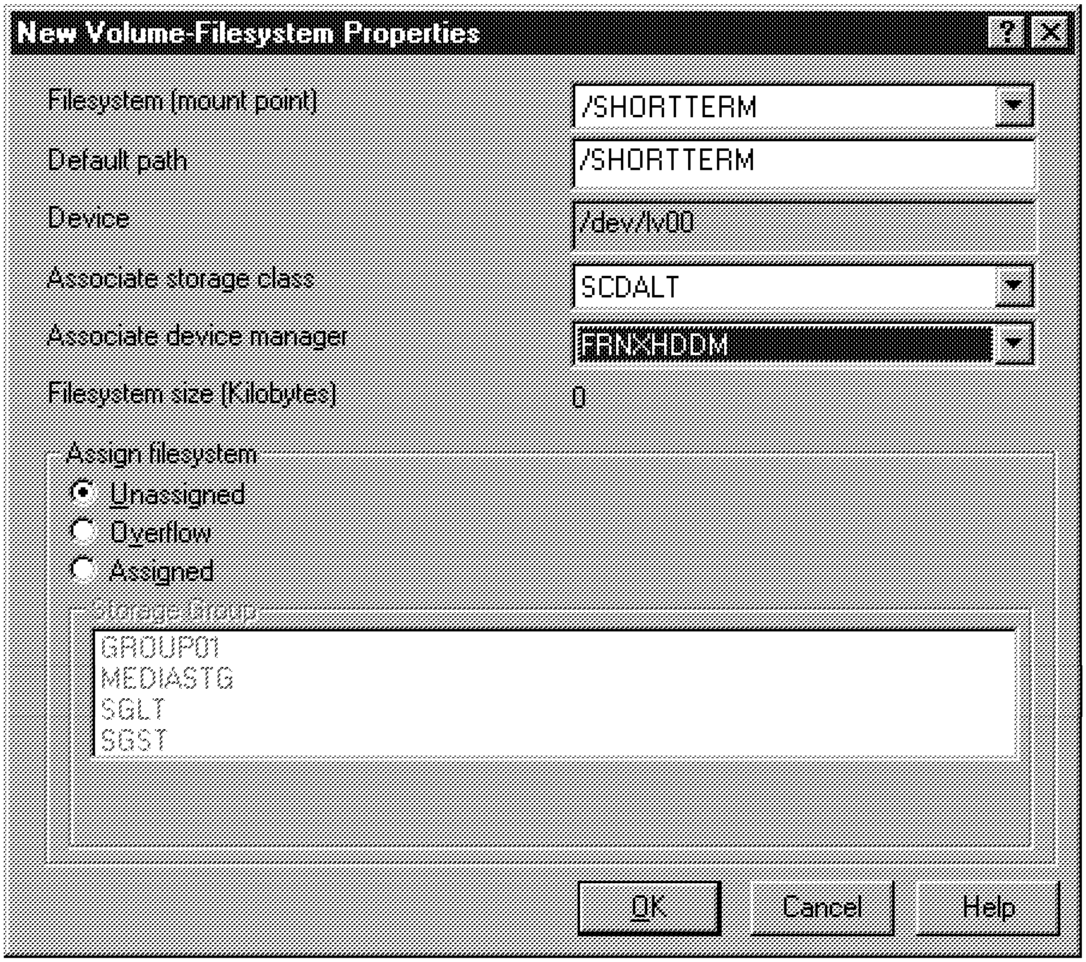
- For a DASD Volume, the name is that of the related file system.
- For an ADSM volume, the name is the concatenation of ADSM with the related management class defined in ADSM.

Return to the SMS components list (Figure 105 on page 148) and double-click on the Volumes folder. The default volume /home appears in the right-hand pane. It does not fit our requirements, so we have to create our own volumes.

To create the first volume on DASD, we click with the right mouse button with the Volumes folder selected and select:

```
 --> New
   --> Filesystem
```

The New Volume-File system properties window appears. Select the appropriate filesystem **/SHORTTERM** (the default path and device are automatically retrieved), its Associate storage class **SCDAST** and Associate device manager **FRNXHDDM**. Check the **Unassigned** option in the Assign filesystem menu. See Figure 110.



New Volume-Filesystem Properties

Filesystem (mount point)	/SHORTTERM
Default path	/SHORTTERM
Device	/dev/lv00
Associate storage class	SCDALT
Associate device manager	FRNXHDDM
Filesystem size (Kilobytes)	0

Assign filesystem

● Unassigned
○ Overflow
○ Assigned

Storage Group

GROUP01
MEDIASTG
SGLT
SGST

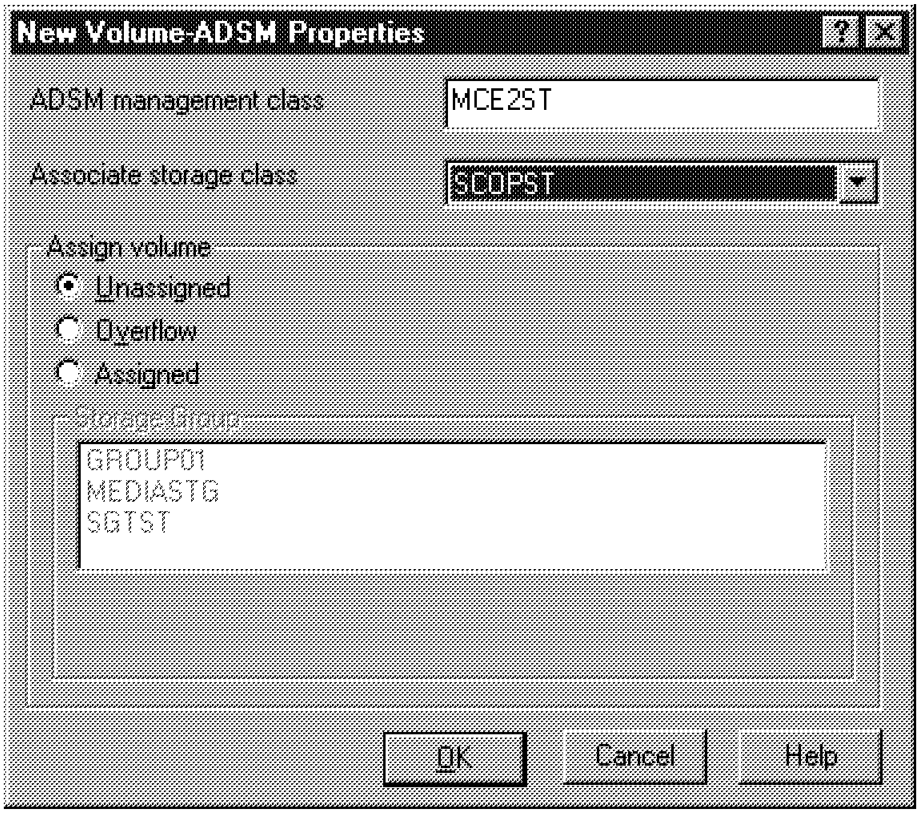
Figure 110. File System Volume Creation

Click on **OK** to register this file system volume.

To create the first volume on ADSM, click on the right mouse button with the **Volumes** folder selected and select:

```
--> New
  --> ADSM
```

The New Volume-ADSM properties window appears. Select the appropriate ADSM management class **MCE2ST** and its Associate storage class **SCOPST**. Check the **Unassigned** option in the Assign volume menu. (See Figure 111 on page 154).



New Volume-ADSM Properties

ADSM management class: MCE2ST

Associate storage class: SCOPST

Assign volume

● Unassigned

○ Overflow

○ Assigned

Storage Group

GROUP01
MEDIASTG
SGTST

Figure 111. ADSM Volume Creation

Click on **OK** to register this ADSM volume. At this time, VisualInfo uses (through the SMS server) the ADSM client installed on ROLM3 to verify that this management class is actually defined in the ADSM server ROLM7. If your system hung after clicking on OK, it could be that your ADSM management class entry is not defined.

Repeat this step to create the other volumes, as specified in our configurations with their appropriate data:

```
Second File system volume --> /LONGTERM
                          --> SCDALT
                          --> FRNXDDM

Second ADSM volume        --> ADSM MCE2LT
                          --> SCOPLT
```

5. Storage Group

A *storage group* is a logical entity that contains storage volumes belonging to different storage classes.

Return to the SMS components list (Figure 105 on page 148) and double-click on the Storage Groups folder. The installed storage groups appear in the right-hand pane. None of them fit our requirements, so we create our own storage group.

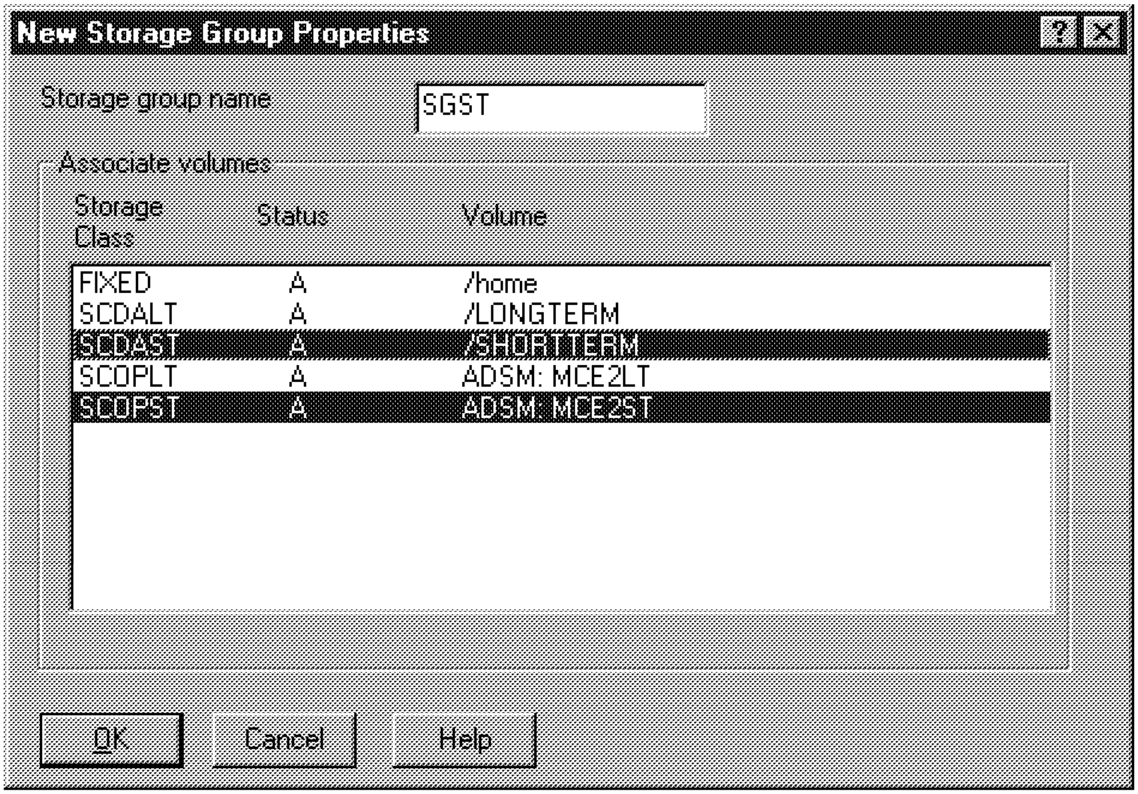
To create the first storage group SGST, click on the right mouse button with Storage Groups folder selected and select:

```
 --> New
```

The New Storage Group Properties window appears. Select its Associate volumes:

/SHORTTERM
/ADSM: MCE2ST

and register its name. (See Figure 112).



Storage group name: SGST

Associate volumes

Storage Class	Status	Volume
FIXED	A	/home
SCDALT	A	/LONGTERM
SCDAST	A	/SHORTTERM
SCOPLT	A	ADSM: MCE2LT
SCOPST	A	ADSM: MCE2ST

Figure 112. Storage Group Creation

Click on **OK** to register this storage group.

Repeat this step to create the other storage group with its Associate volumes:

```
SGLT   --> /LONGTERM
       --> ADSM: MCE2LT
```

6. Collection

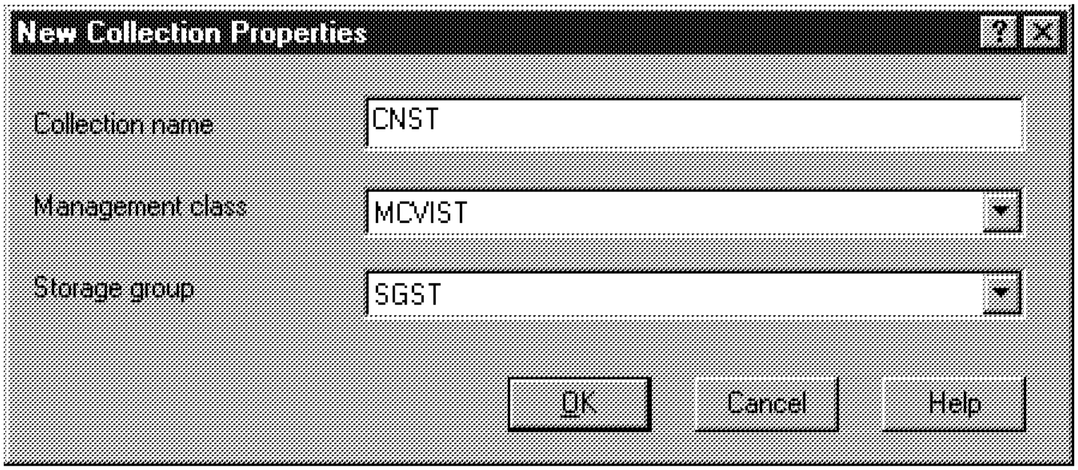
A *collection* consists of the management classes that we associate with storage groups.

Return to the SMS components list (Figure 105 on page 148) and double-click on the Collections folder. The default collections appear in the right-hand pane. None of them fit our requirements, so we create our own collection.

To create the first collection CNST, click on the right mouse button with the Collection folder selected and select:

--> New

The New Collection Properties window appears. Select its appropriate management class **MCVIST** and storage group **SGST**, and then register its name (see Figure 113 on page 156).



New Collection Properties

Collection name: CNST

Management class: MCVIST

Storage group: SGST

Figure 113. Collection Creation

Click on **OK** to register this collection.

Repeat this step to create the other collection with its appropriate data:

```
CNLT   --> MCVILT
       --> SGLT
```

7. Staging area configuration

The *staging area* is the location and amount of space of the temporary storage area that an object server uses before moving objects to SMS storage. By default, it is located inside /ubosstg under the object server administrator directory (/home/viadmin2 in our configuration). We kept this default path.

> **Important**
>
> It is not possible to specify any mount point as a path name for the staging area.

To receive two thousand documents a day (an average document is approximately 100KB), the staging area should be greater than 200MB.

Return to the SMS components list (Figure 105 on page 148) and double-click on the Staging Area folder. The Staging Area Properties window appears. Configure the following entries:

- Maximum size

  Specify how large we want the staging area to be in megabytes. The destager and purger use this size as a threshold to determine when objects should be processed. For our example, we define a staging area of 500MB.

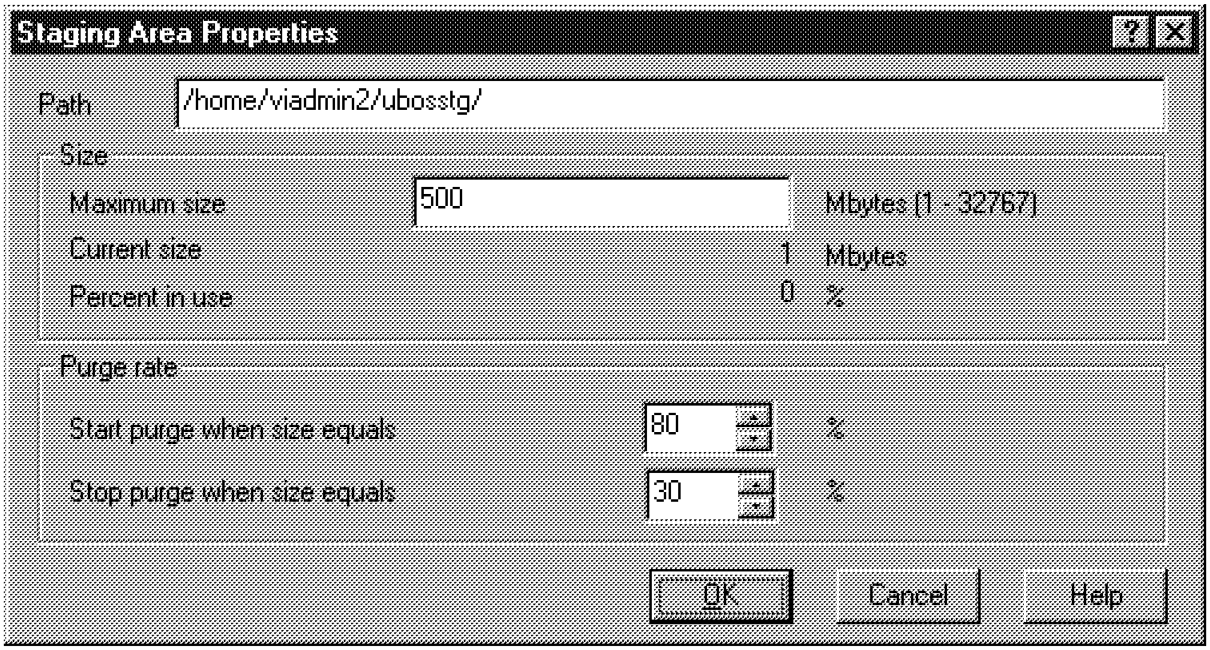
- Start purge when size equals

  Specify the percentage of the staging area that must be in use for the VisualInfo system to start deleting objects from the staging area. For our example, we select a value of 80%.

- Stop purge when size equals

  Specify the percentage of the staging area that must be in use for the VisualInfo system to stop deleting objects from the staging area. For our example, we select a value of 30%.

Figure 114 on page 157 shows this window after we configured these fields.



Staging Area Properties

Path: /home/viadmin2/ubosstg/

Size

Maximum size: 500 Mbytes (1 - 32767)

Current size: 1 Mbytes

Percent in use: 0 %

Purge rate

Start purge when size equals: 80 %

Stop purge when size equals: 30 %

Figure 114. Staging Area Configuration

Click on **OK** to register the staging area configuration.

8. Object server OSR3 Configuration

   To define the configurations for object server OSR3 installed on ROLM3, return to the SMS components list (Figure 105 on page 148) and double-click on the Configuration folder. The default configuration appears in the right-hand pane. It does not fit our requirements, so we create our own configuration, EX2CFG.

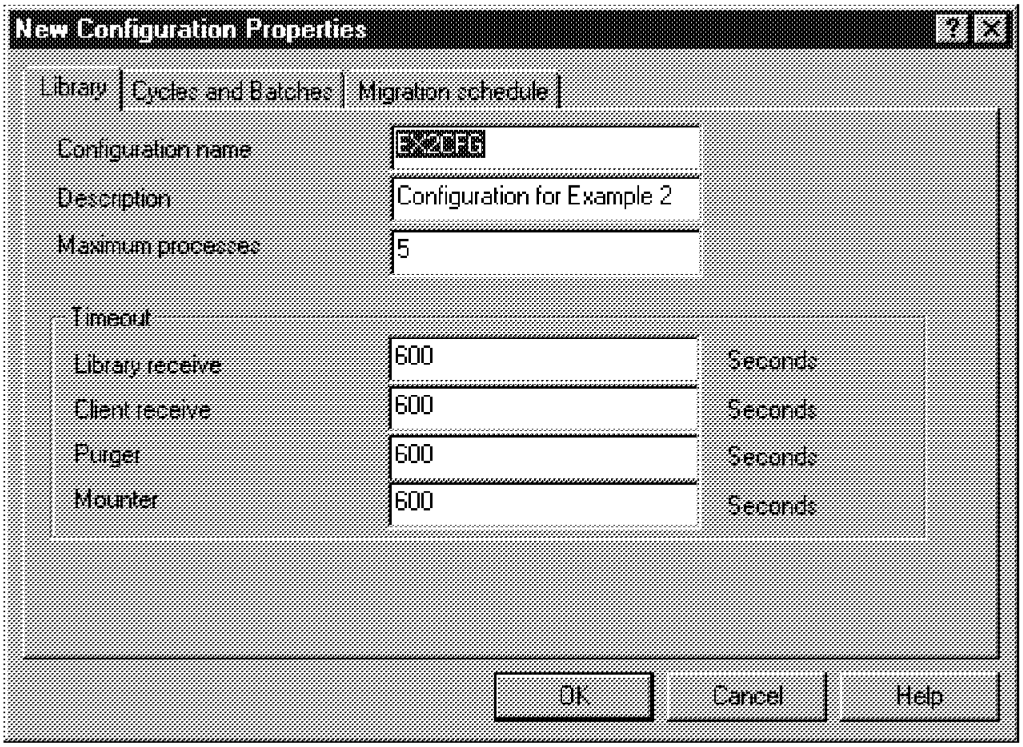
   Click on the right mouse button with the Configuration folder selected, and select:

   --> New

   The New Configuration Properties window appears. Configure the following entries (use the default values for all other entries):

   - Configuration name: EX2CFG
   - Description: Configuration for Example 2.
   - Purger: 30. Thirty minutes must elapse before the system can check and determine whether to delete objects from the staging area, on the base of the purge rate defined for that staging area.
   - Destager: 15. Fifteen minutes must elapse before the system moves the objects from the staging area to the first volume in the primary storage class.
   - Migrator: 15. Fifteen minutes must elapse before the system can migrate objects from one storage class to another.
   - Start time: every day at midnight for eight hours. Specify when the system should move objects between storage classes.

Figure 115 on page 158, Figure 116 on page 158, and Figure 117 on page 159 show the related configuration windows with these entries:



New Configuration Properties

Configuration name: EX2CFG

Description: Configuration for Example 2

Maximum processes: 5

Timeout

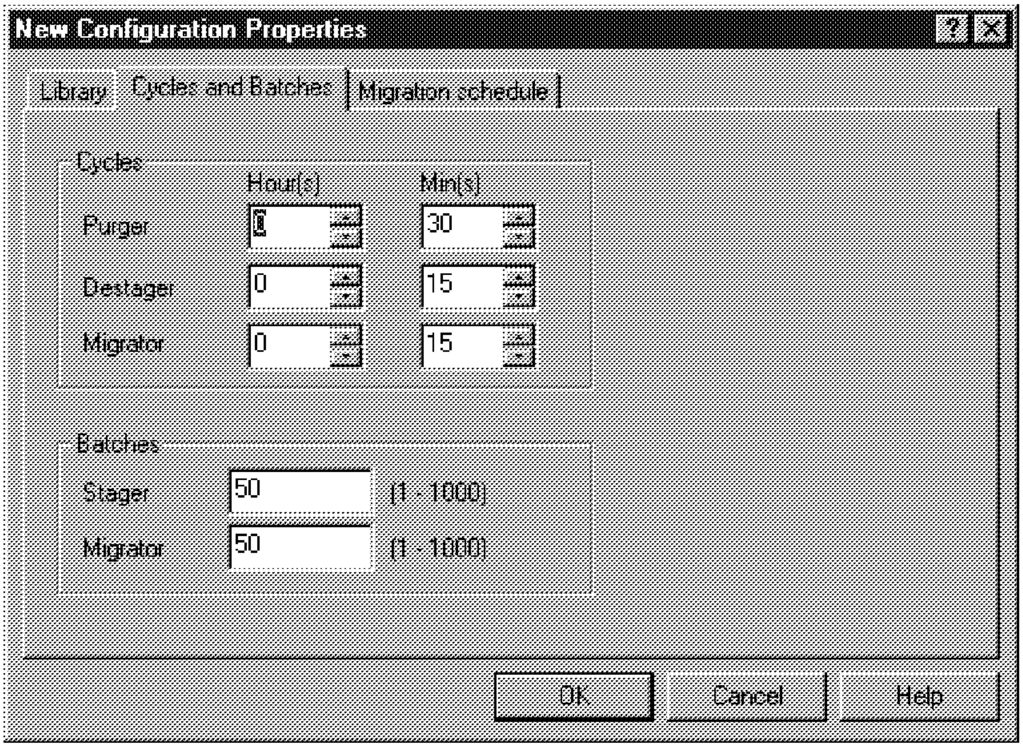
Library receive: 600 Seconds

Client receive: 600 Seconds

Purger: 600 Seconds

Mounter: 600 Seconds

Figure 115. New Configuration Properties-Library



New Configuration Properties

Cycles

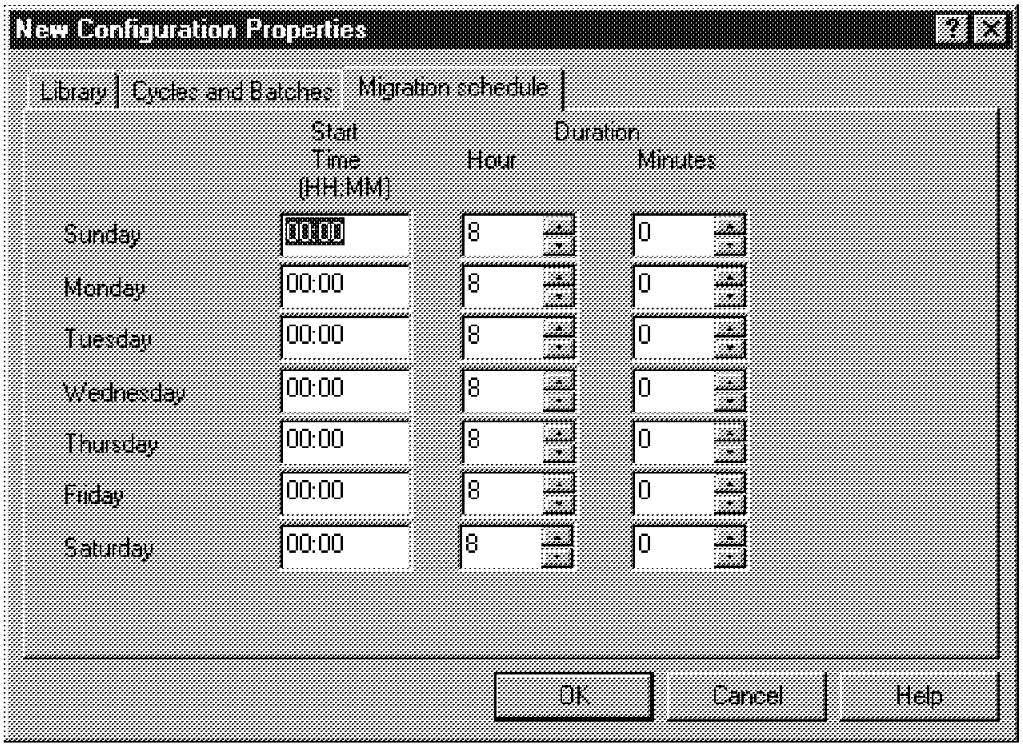
	Hour(s)	Min(s)
Purger	0	30
Destager	0	15
Migrator	0	15

Batches

Stager: 50 (1 - 1000)

Migrator: 50 (1 - 1000)

Figure 116. New Configuration Properties-Cycles and Batches



	Start Time [HH:MM]	Duration Hour	Minutes
Sunday	00:00	8	0
Monday	00:00	8	0
Tuesday	00:00	8	0
Wednesday	00:00	8	0
Thursday	00:00	8	0
Friday	00:00	8	0
Saturday	00:00	8	0

Figure 117. New Configuration Properties-Migration Schedule

Click on **OK** to register the OSR3 configuration.

To have the new object server configuration take effect, execute the following steps on ROLM3:

a. Shut down the object server.

b. Change the FRNDBCFG environment variable in the /home/viadmin2/frnsetup.obj script file to indicate the new configuration to use: EX2CFG.

c. Run the .frnsetup.obj to reset the environment variable for the object server.

d. Restart the object server.

6.4.5.3 Tests

In this section we explain how to test that the environment setup is working according to the customer′s requirements. Here is the test procedure:

1. Import documents inside the VisualInfo platform.
2. Index them using one of the index classes created.
3. Verify that these documents are stored in their first storage class.
4. At the appropriate time, verify that the documents are migrated to their last storage class.

Import Documents: Start the VisualInfo client application on the System Administration workstation and log on library server LSR3. Select the **Import** option as well as the files to import and uncheck the use of workbasket.

When the import operation completes successfully, the files are stored in the ROLM3 staging area, /home/viadmin2/ubosstg, with AIX flag r (read only).

Index Documents: From the VisualInfo client application main menu, select the **Basic Search** option and specify use of the NOINDEX class. Previously imported documents are retrieved and listed. Open and index each of them, using one of the index classes created for this purpose:

- Document for short-term
- Document for long-term

Storage on the First Storage Class: When indexing has completed, each document is sent from the staging area to its first storage media (file systems in our configuration). It stays in the staging area but its related AIX flag is modified to rw (read/write), so that the purger can delete it if space is unavailable in the staging area.

In the /SHORTTERM file system, documents indexed with the Document for short-term index class are registered under (for example) subdirectory ubosdata/00004/00.

In the /LONGTERM file system, documents indexed with the Document for long-term index class are registered under (for example) subdirectory ubosdata/00005/00.

Documents stay in their file system until migration occurs (at midnight in our configuration).

Storage on the Last Storage Class: The next day, check that the documents are erased from the file systems they were stored in and that they are registered inside VI001A (for documents indexed with Document for short-term index class) or VI002A (for documents indexed with Document for short long index class).

To request the contents of each volume, execute the following commands from the ADSM server ADMIN client:

```
adsm> query content VI001A
adsm> query content VI002A
```

Previous documents are stored in these volumes according to their index classes.

This completes our test to determine whether these customers requirements have been fulfilled:

1. Documents are stored successively on different media.
2. Documents with the same lifetime are stored in the same platters.

6.4.5.4 Retrieval Time

Table 6 on page 161 shows the response time we experienced for retrieving a document from its different possible locations in the VisualInfo system.

After being indexed, a document could be found in:

- The staging area
- Magnetic DASD (primary storage class)

- Optical platters inserted in one of the optical drives (secondary storage class)
- Optical platters inserted in the optical library (secondary storage class)

For our test, we worked with a document of 133KB.

Table 6. Response Time for Retrieving Documents

Document Location	**Time**
Document stored in staging area	< 2 seconds
Document stored in DASD	< 2 seconds
Document stored in platters mounted in one of the optical drives	< 3 seconds
Document stored in platters installed in the optical library	< 11 seconds

For the last three configurations tested, we verified that the document was not stored in the staging area.

Note: These response times are given for information only. They are the results of our configuration, and they should not be used as a reference.

Chapter 7. Backing Up and Restoring VisualInfo Data

Once customers have begun to use VisualInfo in production, their main concern is data availability. They want to be able to recover from disasters (for example, disk scratches, floods, fire) as soon as possible with the latest data committed.

In this chapter, we explain which VisualInfo data to back up and the method to use. This example can be used if you are working on NT platforms with the same subsystems, because backup and restore procedures are identical.

We do not cover the backup of the entire server on which VisualInfo is running. We assume that, once a VisualInfo server is ready to use and before any documents are stored, the customer has performed a complete backup (system and subsystems installed and configured).

Because we use the ADSM client on the VisualInfo server to handle migration to an optical library, we decided to use that subsystem to back up and restore VisualInfo data on the ADSM server.

Attention

When purchasing a VisualInfo license, you receive one limited use ADSM license for storing data in or migrating data to an optical library. The use of ADSM for any other purposes (such as backup and archive) require the purchase of a full ADSM license.

7.1 VisualInfo Data to Back Up on AIX and NT

Back up the following VisualInfo data on an AIX or NT platform:

- On the library server
  - Library server database
  - Directory where the index classes DLL are stored
- On the object server
  - Object server database
  - Staging area
  - Directories where the documents are stored

Important

Once one of these resources has been damaged, it should be restored using its backup. But all the resources are dependent, and restoring only one of them could bring inconsistency in the VisualInfo structure. So a detailed study should be run before any restore operation occurs.

There is no procedure to follow for restoring a consistent VisualInfo structure after one of its components fails, but one of these guidelines could be used:

1. Back up all VisualInfo resources at the same time. In case of failure for any one of them, restore all of them at the same time. However, using this solution, all new data since the last backup is lost.
2. Back up all VisualInfo resources at the same time. Then, in case of failure for one of them, retrieve the date and time for the last consistent VisualInfo structure, restore both databases up to this date and time (using the roll forward option), then adapt the directory contents to the database contents. This solution could be difficult to set up if the amount of new data is great.

7.2 DASD Used for VisualInfo Server

To ensure high availability of VisualInfo data, we recommend using the RAID 5 architecture for the disks containing the documents. If a failure occurs on one of the disks defined in the RAID 5 pool, the online operation can still proceed.

Note: Do *not* install the databases on RAID 5 architecture. If you do, performance will be degraded.

The RAID 5 architecture does not remove the need to back up. For example, it cannot restore any documents that are dropped accidentally.

7.3 VisualInfo Directories

The simplest way of backing up data stored in the VisualInfo server directories is to do the following:

1. Configure its ADSM client option file, dsm.opt.
2. Start a client scheduler on the VisualInfo server.
3. Set up a schedule on the ADSM server specifying which directories to back up.
4. Associate this schedule with the target VisualInfo server.

7.4 VisualInfo Databases

Two techniques are available for backing up VisualInfo databases stored in the VisualInfo server:

- Use the DB2 backup utility to back up databases into files, and then use ADSM to back up the image object files. This technique is useful for backing up something smaller than the entire database, such as a table space, or to take advantage of the backup utility that allows online backup. Enough disk space should be available to store the database and its created image file.

- Use ADSM to directly back up the files that make up the database and logs (for example sql0001, sql0002). The advantage of this technique is that no intermediate files are created and the database can be online.

7.5 Backup and Recovery Examples on AIX

In this section, we use the configuration described in Chapter 6, "Setting Up Example 2 on AIX" on page 85 to set up the backup environment for the VisualInfo server.

Remember that:

- ROLM3 is the VisualInfo server with an ADSM client installed.
- ROLM7 is the ADSM server.

7.5.1 Scenario

Every day new documents are indexed and stored on the VisualInfo server, ROLM3. Library server and object server databases are modified, and the contents of file systems /SHORTTERM and /LONGTERM evolve (new documents are stored and migrated documents are deleted).

To prevent the loss of all VisualInfo data if a scratch occurs in the VisualInfo server, we decide to back up the following items every night.

1. At 8 PM, we back up these VisualInfo server directories:
   - Index class DLL directory: /home/viadmin/frn/LSR3/dll (viadmin is the library server database administrator and LSR3 is the library server name)
   - Staging area directory: /home/viadmin2/ubosstg (viadmin2 is the object server database administrator)
   - /SHORTTERM/ubosdata directory
   - /LONGTERM/ubosdata directory
2. At 10 PM, we back up these VisualInfo server databases:
   - Object server database (OSR3)
   - Library server database (LSR3)

The data is backed up on DASD in the ADSM server. This redbook does not cover the archive operation (moving backup data from DASD to tape, for example), which should be defined by the ADSM administrator.

To prevent any user from modifying VisualInfo data during its backup, we select a time frame for this operation when no user is logged on.

7.5.2 Preliminary Steps

Before using ADSM server ROLM7 to back up files and databases from VisualInfo server ROLM3, we had to do the following:

1. Define the management class to use on ADSM
2. Set up environment variables for ADSM
3. Back up the databases offline

7.5.2.1 Management Class Definition

Before starting any backup operation from VisualInfo Server, using its ADSM client, you must create on the ADSM server an environment for the backup requirement.

Note: For this backup, you cannot use any of the management classes already created for a VisualInfo server on an ADSM server to migrate data to an optical library. Instead, Table 7 lists the new ADSM resources to define for this environment.

Table 7. New ADSM Resources to Define for Backup

ADSM Resource	**Name**
Management class	MC4BU
Storage pool	SP4BU
Volume	/dev/rlv09

Because ROLM3 is already registered on the ADSM server in policy domain STANDARD for the VisualInfo migration task, the new management class MC4BU must be registered in this policy domain. Also, the same policy set, PSE2, must be used, because only one policy domain can be active at a time and all the management classes should be used at the same time.

Execute the following ADSM commands from an ADSM administrative client interface to set up this new ADSM environment:

- Management class MC4BU definition

  Run the following command to create management class MC4BU:

  ```
  adsm> define mgmtclass STANDARD PSE2 MC4BU
  ```

- Default management class definition

  Run the following command to define management class MC4BU as the default one.

  ```
  adsm> assign defmgmtclass STANDARD PSE2 MC4BU
  ```

  After this, it will not be necessary to specify the management class to use for each backup request.

- Copy group definition

  Run the following command to create a copy group inside management class MC4BU and associate it with the new storage pool:

  ```
  adsm> define copygroup STANDARD PSE2 MC4BU destination=SP4BU
  ```

- Storage pool definition

  Run the following command to create storage pool SP4BU:

  ```
  adsm> define stgpool SP4BU DISK
  ```

- Activate policy set

  Run the following command to activate policy set PSE2:

  ```
  adsm> activate policyset STANDARD PSE2
  ```

- Volume definition

Create file system FS4BU on the ADSM server to store the backup data. Its logical volume (raw device name), /dev/rlv09, is used to create the related ADSM volume.

Run the following command to create this /dev/rlv09 ADSM volume and associate it with the new storage pool SP4BU.

```
adsm> define volume SP4BU /dev/rlv09
```

Important

Files and databases backed up with this new management class are copied from the ADSM client to the ADSM server on file system /FS4BU. Because the ADSM server uses the logical volume raw device of this file system to store the backup data, it is not possible to list files and images databases backed up by running the following AIX command:

```
# ls -l /FS4BU
```

To verify that the files and images database are actually backed up, use the following ADSM command, which lists the content of the target volume used to store backup data:

```
adsm> query content /dev/rlv09
```

7.5.2.2 Set Up Environment Variables for ADSM

Before the database manager can use the ADSM option to back up and restore databases, the following environment variables should be defined for the instance owner ID vidb2in:

- DSMI_DIR: it identifies the directory path where the API trusted agent file (dsmapita) and the API message files (dscameng.txt and dsgameng.txt) are located. For our example we selected the default path /usr/lpp/adsm/bin.
- DSMI_CONFIG: it identifies the directory path and file name containig the ADSM user options. For our example we selected the default file /usr/lpp/adsm/bin/dsm.opt.
- DSMI_LOG: it identifies the directory path where the error log (*dsierror.log*) will be created. For our example, we selected the */* directory.

For this purpose, include the following lines in the profile of user ID vidb2in (/home/vidb2in/.profile).

```
export DSMI_DIR=/usr/lpp/adsm/bin
export DSMI_CONFIG=/usr/lpp/adsm/bin/dsm.opt
export DSMI_LOG=/home/vidb2in
```

If the database manager is running, first stop it (db2stop) then start it (db2start), so that these variables will take effect.

7.5.2.3 Offline Database Backup

To back up VisualInfo databases, we used the online backup option provided by the DB2/6000 backup utility. There is no need to stop applications using these databases during the backup operation.

However, you cannot perform online database backup if the database is not logging for forward recovery, as this log will be required at restore time. Follow these steps to initiate forward recovery for database LSR3 on the VisualInfo server:

1. Set the log retain for the LSR3 database.

   Log on to the VisualInfo server, using the database instance vidb2in owner ID (vidb2in), and start the database administrative GUI:

   ```
   $ db2dd
   ```

   > **Attention**
   >
   > If problems occur when you run this command, it could be that the PATH variable is not set correctly for the database instance owner. If so, add in the vidb2in user ID .profile file the following line:
   > . /home/vidb2in/sqllib/db2profile

   The Database Director - Tree View window is displayed (see Figure 118).

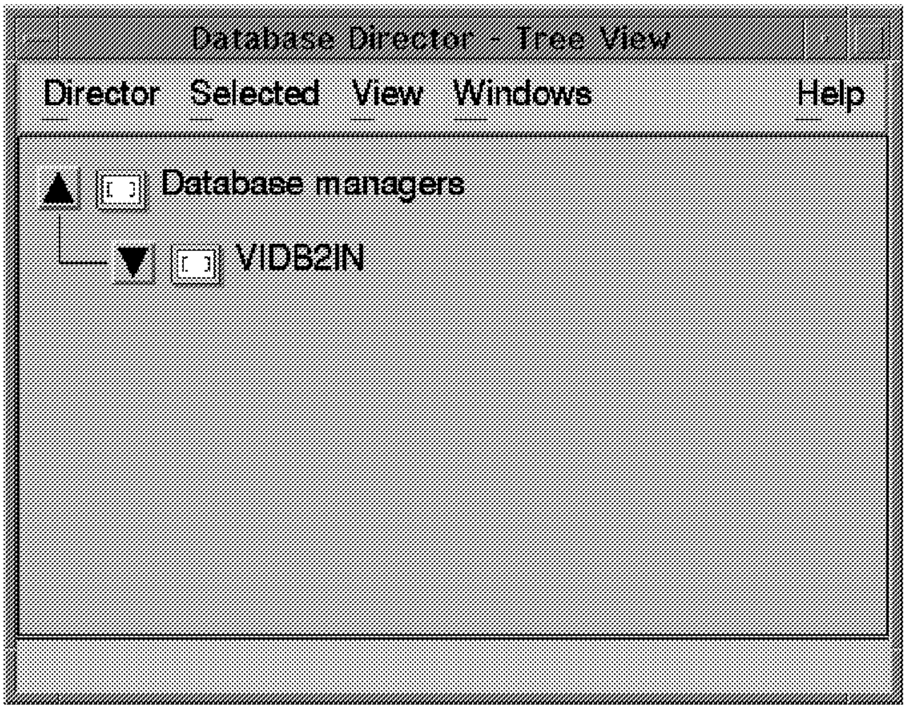


Figure 118. Database Director: Instance Level

   Select VisualInfo VIDB2IN instance, where the LSR3 database has been recorded, and click on its black arrow icon. The instance subdirectories is shown (see Figure 119 on page 169).

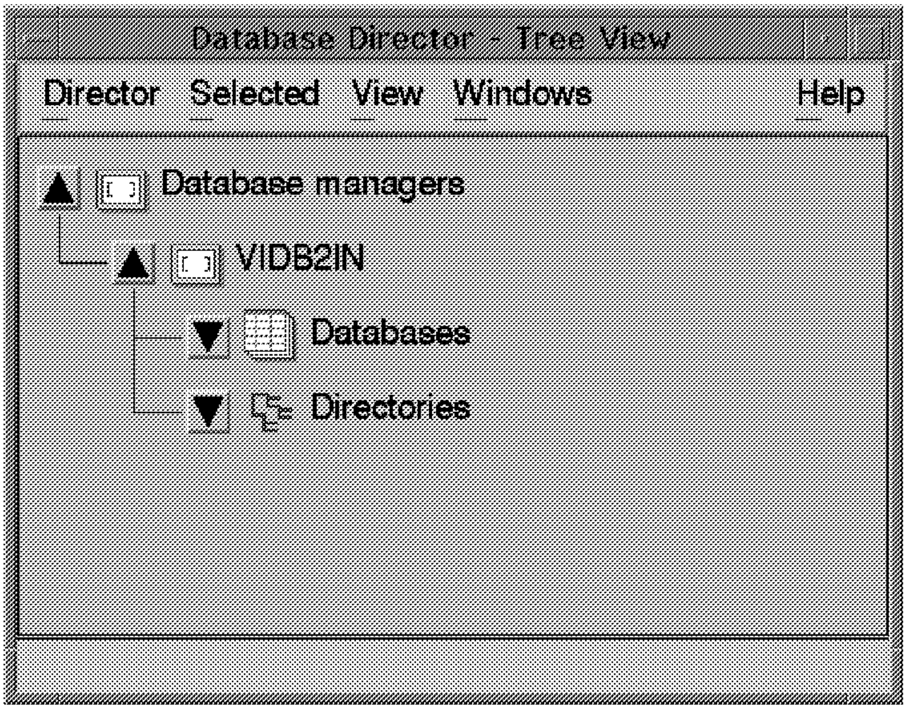


Figure 119. Database Director: Instance Subdirectories

Select **Databases** and click on its black arrow icon. The databases recorded in the VisualInfo VIDB2IN instance are displayed (see Figure 120).

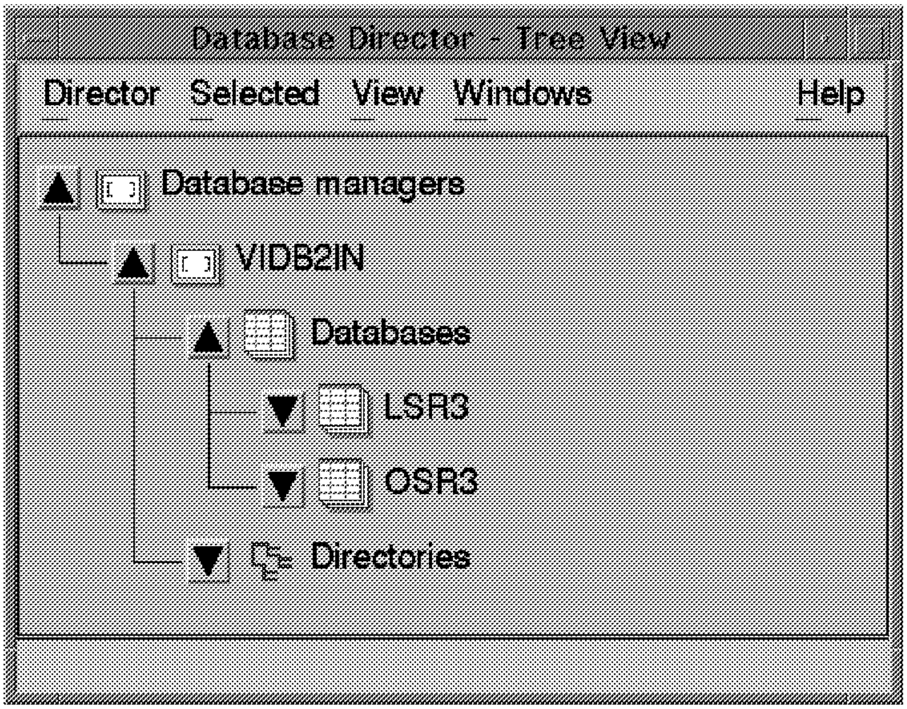
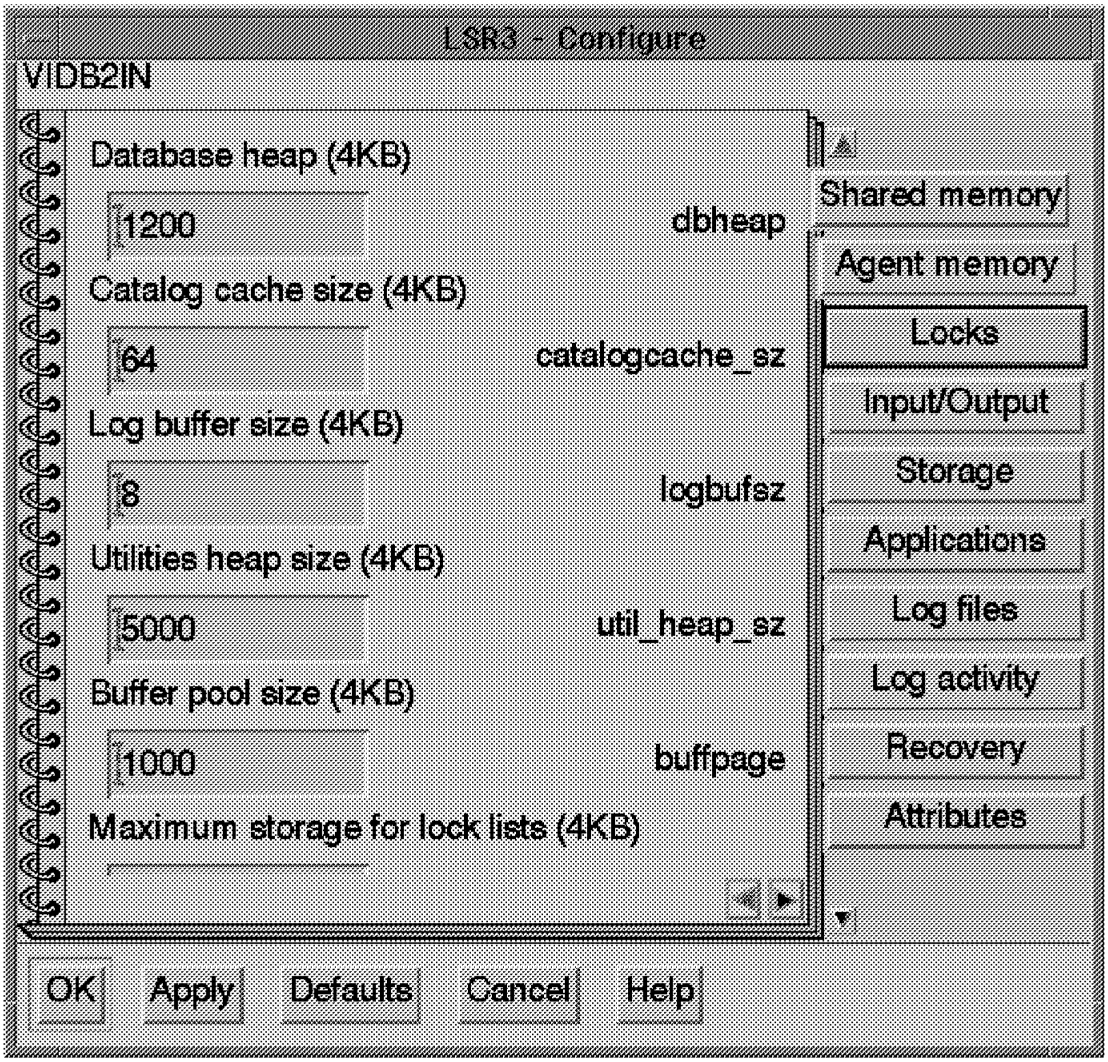


Figure 120. Database Director: Databases

Select **LSR3** and click on it with the right-hand mouse button. On the pop-up menu, select the **configuration** option to display the LSR3 - Configure window (see Figure 121 on page 170).



LSR3 - Configure

VIDB2IN

Database heap (4KB)

1200 dbheap

Catalog cache size (4KB)

64 catalogcache_sz

Log buffer size (4KB)

8 logbufsz

Utilities heap size (4KB)

5000 util_heap_sz

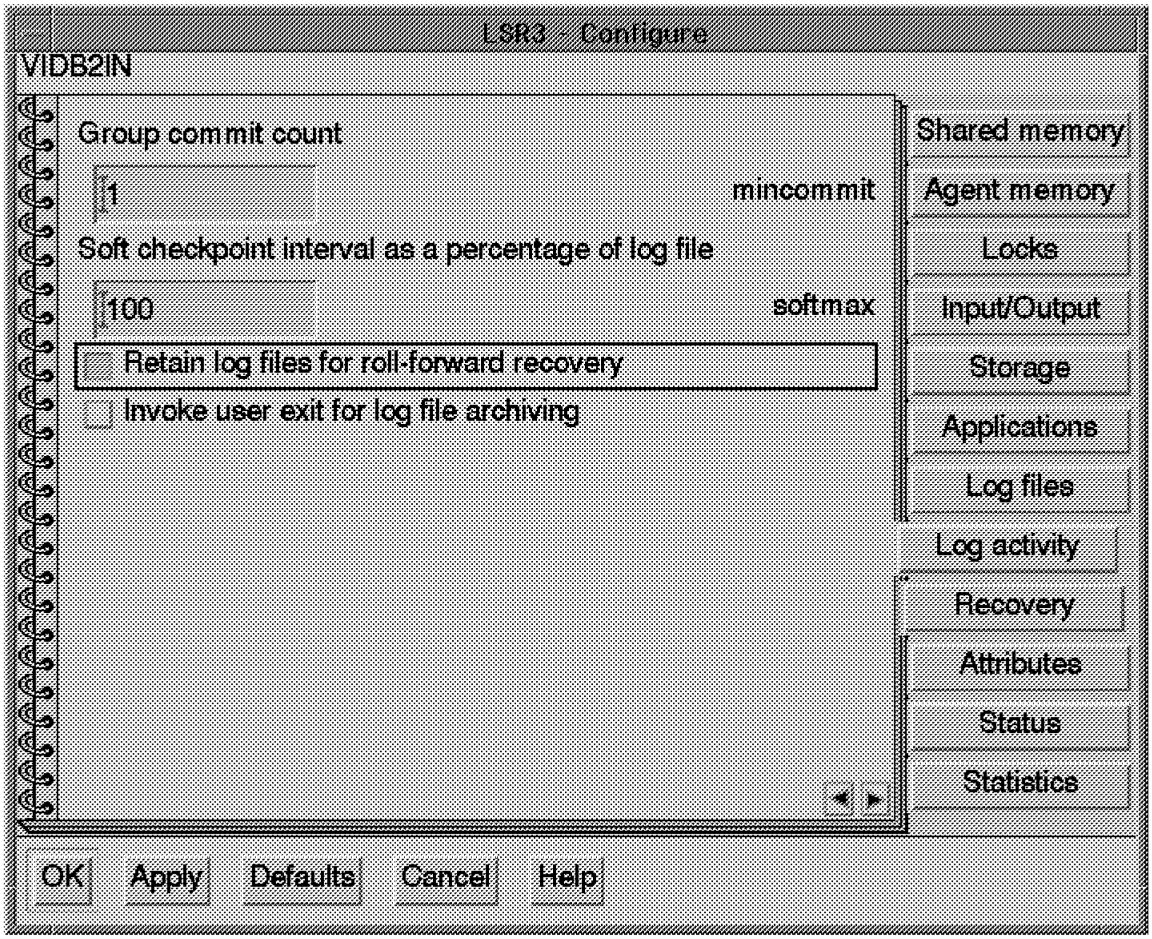
Buffer pool size (4KB)

1000 buffpage

Maximum storage for lock lists (4KB)

Figure 121. LSR3 - Configure Window

Click on **Log activity** to display the Log activity Configuration menu (see Figure 122 on page 171).



Group commit count

1 mincommit

Soft checkpoint interval as a percentage of log file

100 softmax

□ Retain log files for roll-forward recovery

□ Invoke user exit for log file archiving

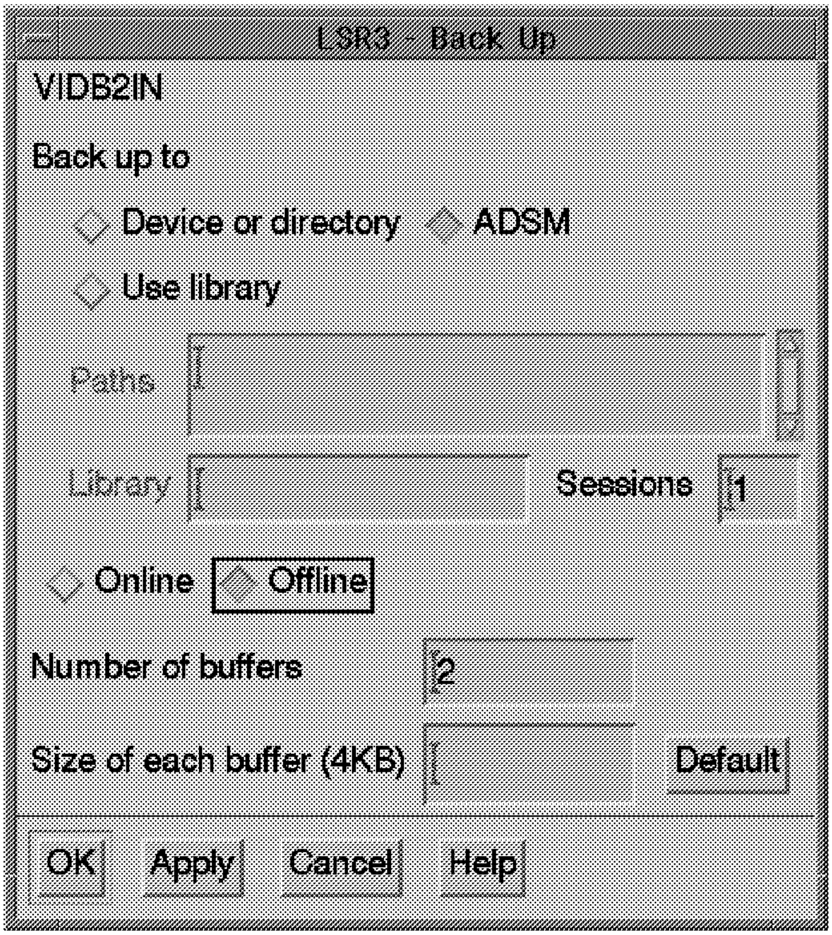
Figure 122. LSR3 - Configure Window

To enable forward recovery, select the Retain log files for roll-forward recovery option and save the new LSR3 configuration.

2. Perform LSR3 offline backup.

   Once the log retain parameter has been set, an offline database is required to use the LSR3 database.

   Return to the Database Director: Databases window (see Figure 120 on page 169) and select **LSR3**. Click on it with the right-hand mouse button. On the pop-up menu select the **back up** option to get the LSR3 - Back Up window (see Figure 123 on page 172).



LSR3 - Back Up

VIDB2IN

Back up to

◇ Device or directory ◆ ADSM

◇ Use library

Paths

Library

Sessions 1

◇ Online ◆ Offline

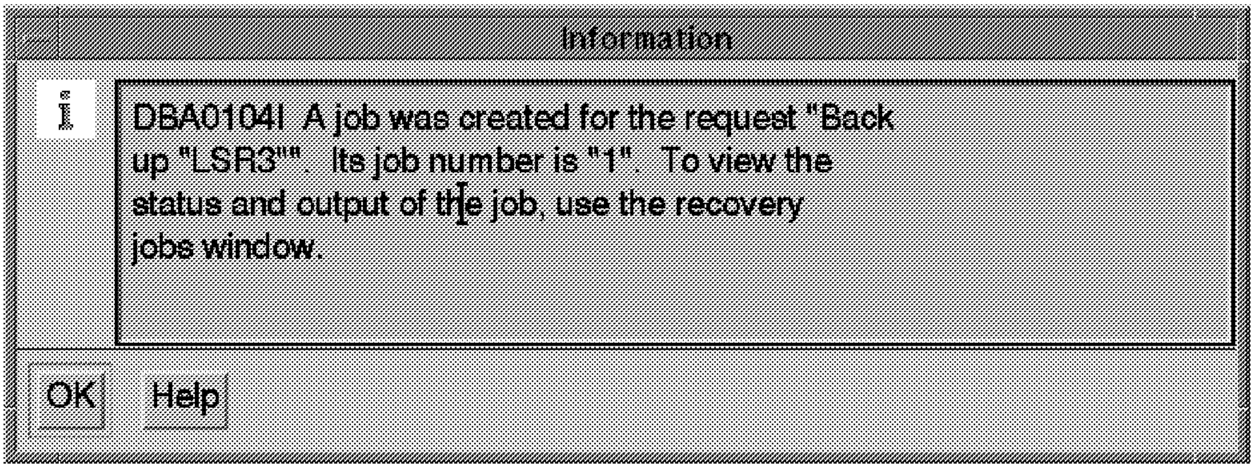
Number of buffers 2

Size of each buffer (4KB)

Figure 123. LSR3 - Back Up

Select **Back up to ADSM** and **Offline**.

To start the offline LSR3 database backup, click on **OK**. An information window is displayed (see Figure 124).

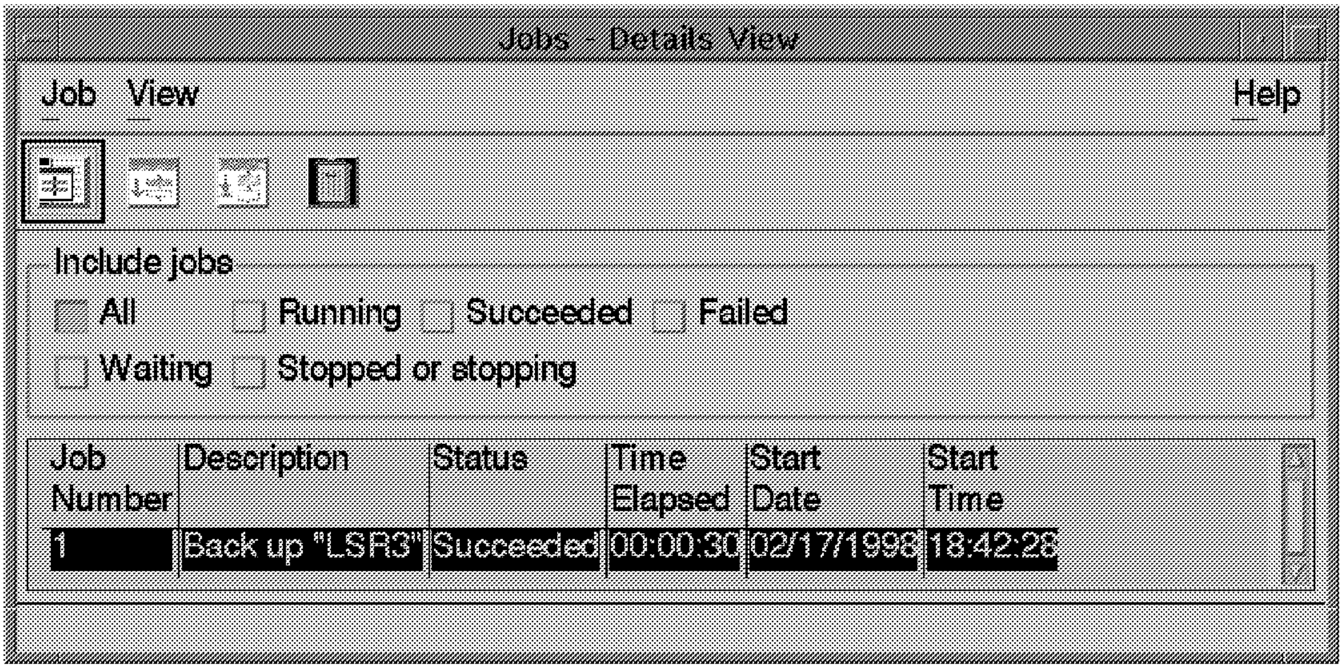


Information

DBA0104I A job was created for the request "Back up "LSR3"". Its job number is "1". To view the status and output of the job, use the recovery jobs window.

Figure 124. Backup Job Number Information

To verify the status of this backup operation, return to the Database Director: Databases windows (see Figure 120 on page 169) and select **Windows** from the menu bar. On the pop-up select **Recovery Status** to get the Jobs - Details View window (see Figure 125 on page 173).



Include jobs

☒ All ☐ Running ☐ Succeeded ☐ Failed

☐ Waiting ☐ Stopped or stopping

Job Number	Description	Status	Time Elapsed	Start Date	Start Time
1	Back up "LSR3"	Succeeded	00:00:30	02/17/1998	18:42:28

Figure 125. Jobs - Details View Window

Verify that the status of Job Number 1 is set to Succeeded.

Run the procedure described in 7.5.2.3, "Offline Database Backup" on page 167 for the OSR3 database to enable its online backup.

7.5.3 Directory Backup

In this section we present the steps to back up the following VisualInfo server ROLM3 directories from ADSM server ROLM7 every night at 8 PM:

- /SHORTTERM/ubosdata
- /LONGTERM/ubosdata
- /home/viadmin2/ubosstg
- /home/viadmin/frn/LSR3/dll

1. Start the ADSM client scheduler on the VisualInfo server by executing the following AIX command:

   ```
   nohup dsmc schedule 2> /dev/null &
   ```

   The client scheduler runs in the background and keeps running even when the user logs off.

   If you want to start the client scheduler with ROLM3, insert the following line in the /etc/inittab file:

   ```
   adsm::once:/usr/lpp/adsm/bin/dsmc sched >/dev/null 2>1& # ADSM
   scheduler
   ```

   Because the default management class MC4BU fits our requirement for backing up VisualInfo server data, it is not necessary to specify the management class for each directory to back up.

2. Edit the VisualInfo server /usr/lpp/adsm/bin/dsm.opt file and add the following line to allow subdirectories of the directories specified previously to be backed up.

   ```
   SUBDIR  Yes
   ```

3. Set up a central schedule, BU_VI_SRV, on the ADSM server to back up the selected VisualInfo server directories every night at 8 PM. To create this schedule, run the following ADSM command:

   ```
   adsm> define schedule STANDARD BU_VI_SRV description=″Daily back up
   for VisualInfo server ROLM3″ starttime=20:00 period=1 peruits=days
   dayofweek=weekday objects=″/SHORTTERM/ubosdata/* /LONGTERM/ubosdata/*
   /home/viadmin2/ubosstg/* /home/viadmin/frn/LSR3/dll/*″ action=incremental
   ```

4. Associate the VisualInfo server ROLM3 with this schedule by running the following ADSM command:

   ```
   adsm> define association STANDARD BU_VI_SRV ROLM3
   ```

7.5.4 Database Backup

To back up the library server LSR3 and object server OSR3 databases installed on ROLM3 from ADSM server ROLM7 every night at 10 PM, follow these steps:

1. On the VisualInfo server, create a command file that contains the requested AIX commands for backing up LSR3 and OSR3 databases. Figure 126 shows the file created, /home/vidb2in/dbe2bu.

```
db2 connect to LSR3
db2 back up database LSR3 online use adsm
db2 connect to OSR3
db2 back up database OSR3 online use adsm
```

Figure 126. AIX Commands to Back Up LSR3 and OSR3 Databases

   The option *use adsm* allows the backup utility to route the backup image to the default ADSM management class.

   When the ADSM server schedules the execution of this command file on an ADSM client, the root user ID is used to run it on this target client. To allow root ID to execute the command file /home/vidb2in/dbe2bu, run the following commands:
   `chown root /home/vidb2in/dbe2bu` (root becomes the owner of this file)
   `chmod 744 /home/vidb2in/dbe2bu` (root can execute this AIX file, but *not* necessarily its content)

   The file contains DB2 commands that can be run only by users belonging to the db2admin group. So you must do both of the following:

   - Include the root user ID in the db2admin group.
   - Include . /home/vidb2in/sqllib/db2profile in root .profile. This sets up the necessary environment variables to execute DB2 commands.

2. Set up a central schedule, BU_DB_VI_SRV, on the ADSM server to execute every night at 10 PM on the remote ADSM client the /home/viadmin2/dbe2bu command file. To create this schedule, run the following ADSM command:

   ```
   adsm> define schedule STANDARD BU_DB_VI_SRV description=″Daily database
   back up for VisualInfo server ROLM3″ starttime=22:00 period=1 peruits=days
   dayofweek=weekday action=command objects=″/home/viadmin2/dbe2bu″
   ```

3. Associate the VisualInfo server ROLM3 with this schedule. Run the following ADSM command:

   ```
   adsm> define association STANDARD BU_DB_VI_SRV ROLM3
   ```

7.5.5 Tests

The simplest way to verify that the schedule configurations are set correctly is to start both the central schedule and client scheduler. The next morning, verify that these operations have completed successfully by editing the scheduler log file dsmsched.log.

The log file that is stored in the directory from the client scheduler is executed. Figure 127 shows the main part of our schedule report.

```
02/19/1998 18:32:31 Node Name: ROLM3
02/19/1998 18:32:31 Session established with server ADSM: AIX-RS/6000
02/19/1998 18:32:31  Server Version 3, Release 1, Level 0.12
02/19/1998 18:32:31  Server date/time: 02/19/1998 18:22:03  Last access:
02/19/1998 18:21:26

02/19/1998 18:32:31 Querying server for next scheduled event.
02/19/1998 18:32:31 Next operation scheduled:
02/19/1998 18:32:31 ------------------------------------------------------
02/19/1998 18:32:31 Schedule Name:         BU_VI_SRV
02/19/1998 18:32:31 Action:                Incremental
02/19/1998 18:32:31 Objects:               /SHORTTERM/* /LONGTERM/*
/home/viadmin/frn/LSR3/dll/* /home/viadmin2/ubosstg/*
02/19/1998 18:32:31 Options:
02/19/1998 18:32:31 Server Window Start:   20:00:00 on 02/19/1998
02/19/1998 18:32:31 ------------------------------------------------------
02/19/1998 18:32:31 Command will be executed in 1 hour and 38 minutes.
02/19/1998 18:32:31
Time remaining until execution:
Executing scheduled command now.
02/19/1998 20:10:31 Node Name: ROLM3
02/19/1998 20:10:31 Session established with server ADSM: AIX-RS/6000
02/19/1998 20:10:31   Server Version 3, Release 1, Level 0.12
02/19/1998 20:10:31   Server date/time: 02/19/1998 20:00:03  Last access:

02/19/1998 20:10:31 Incremental back up of volume ′/SHORTTERM/*′
02/19/1998 20:10:31 Normal File-->            5,705 /SHORTTERM/ubosdata
/00004/00/FAFMGBOD.HBL ·Sent“
02/19/1998 20:10:31 Normal File-->            5,800 /SHORTTERM/ubosdata
/00004/00/KADOOALM.OGO ·Sent“
etc ...

02/19/1998 20:10:31 Successful incremental back up of′/SHORTTERM/*′
1
02/19/1998 20:10:31 Incremental back up of volume ′/LONGTERM/*′
02/19/1998 20:10:31 Normal File-->            6,183 /LONGTERM/ubosdata
/00005/00/BCLHCFAE.FPK ·Sent“
02/19/1998 20:10:31 Normal File-->            6,563 /LONGTERM/ubosdata
/00005/00/CCGJAGFB.GBO ·Sent“
etc ...

02/19/1998 20:10:31 Successful incremental back up of ′/LONGTERM/*′
2
```

Figure 127 (Part 1 of 3). Schedule Report

```
02/19/1998 20:10:31 Incremental back up of volume '/home/viadmin/frn/LSR3
/dll/*'
02/19/1998 20:10:31 Successful incremental back up of
'/home/viadmin/frn/LSR3/dll/*' 3

02/19/1998 20:10:31 Incremental back up of volume '/home/viadmin2/ubosstg
/*'
02/19/1998 20:10:31 Expiring-->                5,977 /home/viadmin2
/ubosstg/00001.BX1JJSRJ.C@N#9#XO.FRN$NULL.P1.VO ·Sent“
02/19/1998 20:10:31 Normal File-->             5,800 /home/viadmin2
/ubosstg/00001.C8LEEEEI.XIZJIKPT.FRN$NULL.P1.VO ·Sent“
02/19/1998 20:10:31 Expiring-->                1,004 /home/viadmin2
/ubosstg/00001.Q1BO$SXG.A6H433B9.FRN$NULL.P1.VO ·Sent“
etc ..

02/19/1998 20:10:32 Successful incremental back up of
'/home/viadmin2/ubosstg/* 4

02/19/1998 20:10:32 Total number of objects inspected:       97
02/19/1998 20:10:32 Total number of objects backed up:       10
02/19/1998 20:10:32 Total number of objects updated:          0
02/19/1998 20:10:32 Total number of objects rebound:          0
02/19/1998 20:10:32 Total number of objects deleted:         13
02/19/1998 20:10:32 Total number of objects failed:           0
02/19/1998 20:10:32 Total number of bytes transferred:    49.95 KB
02/19/1998 20:10:32 Data transfer time:                    0.00 sec
02/19/1998 20:10:32 Network data transfer rate:        53,253.88 KB/sec
02/19/1998 20:10:32 Aggregate data transfer rate:         61.99 KB/sec
02/19/1998 20:10:32 Objects compressed by:                    0%
02/19/1998 20:10:32 Elapsed processing time:           00:00:00
02/19/1998 20:10:32 Scheduled event 'BU_VI_SRV'
completed successfully. 5
02/19/1998 20:10:32 Sending results for scheduled event 'BU_VI_SRV'.
02/19/1998 20:10:32 Results sent to server for scheduled event 'BU_VI_SRV'.
02/19/1998 20:10:32 Node Name: ROLM3
02/19/1998 20:10:32 Session established with server ADSM: AIX-RS/6000
02/19/1998 20:10:32   Server Version 3, Release 1, Level 0.12
02/19/1998 20:10:32   Server date/time: 02/19/1998 20:00:04  Last access:
02/19/1998 20:00:03

02/19/1998 20:10:32 Querying server for next scheduled event.
02/19/1998 20:10:32 Next operation scheduled:
02/19/1998 20:10:32 ------------------------------------------------------
02/19/1998 20:10:32 Schedule Name:         BU_DB_VI_SRV
02/19/1998 20:10:32 Action:                Command
02/19/1998 20:10:32 Objects:               /home/vidb2in/dbe2bu
02/19/1998 20:10:32 Options:
02/19/1998 20:10:32 Server Window Start:   22:00:00 on 02/19/1998
02/19/1998 20:10:32 ------------------------------------------------------
02/19/1998 20:10:32 Command will be executed in 2 hours.
02/19/1998 20:10:32
Time remaining until execution:
Executing scheduled command now.
02/19/1998 22:10:32 Node Name: ROLM3
02/19/1998 22:10:32 Session established with server ADSM: AIX-RS/6000
02/19/1998 22:10:32   Server Version 3, Release 1, Level 0.12
02/19/1998 22:10:32   Server date/time: 02/19/1998 22:00:04  Last access:
02/19/1998 20:00:04

02/19/1998 22:10:32
Executing Operating System command or script:
   /home/vidb2in/dbe2bu
02/19/1998 22:11:24 Finished command.  Return code is:  0
02/19/1998 22:11:24 Scheduled event
'BU_DB_VI_SRV' completed successfully. 6
02/19/1998 22:11:24 Sending results for scheduled event 'BU_DB_VI_SRV'.
```

Figure 127 (Part 2 of 3). Schedule Report

```
02/19/1998 22:11:24 Node Name: ROLM3
02/19/1998 22:11:24 Session established with server ADSM: AIX-RS/6000
02/19/1998 22:11:24   Server Version 2, Release 1, Level 0.12
02/19/1998 22:11:24   Server date/time: 02/19/1998 22:00:56  Last access:
02/19/1998 22:00:32

02/19/1998 22:11:24 Results sent to server for scheduled event
'BU_DB_VI_SRV'

02/19/1998 22:11:24 Node Name: ROLM3
02/19/1998 22:11:24 Session established with server ADSM: AIX-RS/6000
02/19/1998 22:11:24   Server Version 2, Release 1, Level 0.12
02/19/1998 22:11:24   Server date/time: 02/19/1998 22:00:56  Last access:
02/19/1998 22:00:56

02/19/1998 22:11:24 Querying server for next scheduled event.
02/19/1998 22:11:24 Next operation scheduled:
02/19/1998 22:11:24 -------------------------------------------------------
02/19/1998 22:11:24 Schedule Name:         BU_VI_SRV       02/19/1998
02/19/1998 22:11:24 Action:                Incremental
02/19/1998 22:11:24 Objects:               /SHORTTERM/* /LONGTERM/*
/home/viadmin/frn/LSR3/dll/* /home/viadmin2/ubosstg/*
02/19/1998 22:11:24 Options:
02/19/1998 22:11:24 Server Window Start:   20:00:00 on 02/20/1998
02/19/1998 22:11:24 -------------------------------------------------------
02/19/1998 22:11:24 Schedule will be refreshed in 12 hours.
02/19/1998 22:11:24
```

Figure 127 (Part 3 of 3). Schedule Report

1, 2, 3 and 4 show the backup status for each selected directory.

5 shows the status of the BU_VI_SRV schedule.

6 shows the status of the BU_DB_VI_SRV schedule.

When we checked the content of volume /dev/rlv09, we found all backup files but we could not find the LSR3 or OSR3 database backup.

The database backup operation status returned by DB2/6000 indicated successful completion, but the storage of the image backup actually failed because was not enough space was available on the target filesystem /FS4BU.

We used the ADSM client administrator command `query actlog` to obtain the reason for the failure. ADSM generates this report.

The backup of the databases actually did succeed, but the storage of the database images on ADSM server failed. DB2/6000 should test the return code provided by ADSM before announcing database backup successful.

Important: Until this inconsistency is corrected, always query the content of the ADSM volume to verify that databases are actually backed up.

To proceed with the databases backup steps, we created a new AIX file system, /FSFBU, and added its related logical volume /dev/rlv10, as a new volume to the storage pool SP4BU used for backing up data.

```
Node Name   Type     Filespace      Client's Name for File
                       Name
--------    ----     ----------     -------------------------------
ROLM3       Bkup     /home          /viadmin2/ubosstg/
                                    00001.C8LEEEEI.XIZJKPT.FRN$NULL.P1.-
                                    V0
ROLM3       Bkup     /home          /viadmin2/ubosstg/
                                    00001.G1TT##@Z.LI1LKMR7.FRN$NULL.P1.-
                                    V0
ROLM3       Bkup     /home          /viadmin2/ubosstg/
                                    00001.G9XTZJE#.SYWGFHC8.FRN$NULL.P1.-
                                    V0
ROLM3       Bkup     /home          /viadmin2/ubosstg/
                                    00001.GW8NKZKQ.NRVFEGB7.FRN$NULL.P1.-
                                    V0
etc ...
```

Figure 128. Files Backed Up and Stored on /SP4BU Volume /dev/rlv09

Figure 128 and Figure 129 show the contents of /SP4BU when this volume has been added.

```
Node Name   Type     Filespace      Client's Name for File
                       Name
--------    ----     ----------     -------------------------------
ROLM3       Bkup     /LSR3          /NODE_0/ FULL_BACKUP.19980220181912.1
ROLM3       Bkup     /OSR3          /NODE_0/ FULL_BACKUP.19980220181937.1
```

Figure 129. Databases Backed Up and Stored on /SP4BU Volume /dev/rlv10

The file name of the image databases backed up and stored on the ADSM server contains the time stamp of the database backup operation.

7.5.6 Recovery Example

In this section we describe the steps for recovering VisualInfo data after a failure in the object server database.

In the configuration used, VisualInfo databases are backed up every night at 10 PM and VisualInfo directories are backed up every night at 8 PM. To simulate a database failure, we delete the contents of table viadim2.base_objects of the object server database OSR3.

The result is seen from the VisualInfo client application; when trying to retrieve documents we got the following message:

```
FRN6056A: A library function error 0 occurred. SQL code or internal
order code=I>>J. The system cannot convert the error folder return code.
```

To restore VisualInfo data in a consistent way, we decided to do the following:

- Restore both of its databases from the last backup (the previous night at 10 PM) and apply the roll forward with the date and time of the backup. Then no transactions logged will be applied.
- Restore its directories from the last backup.

Note: Again, this is not a general procedure to apply to every configuration.

1. Object server database OSR3 recovery
    - Stop the object server application, using the viadmin2 user ID.

- Start the database administration application by executing the following command, using the root user ID (root profile has been modified to run this application).

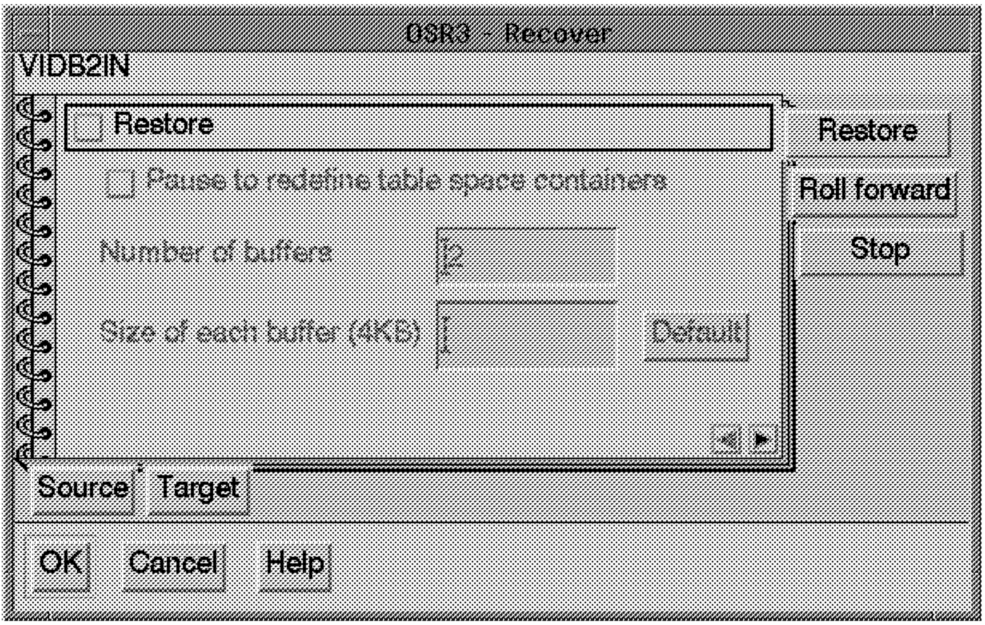
```
# db2dd
```

- Configure the restore operation.

  The Database Director-Tree View window is displayed (see Figure 118 on page 168). Select successively: **VIDB2IN Databases**

  Figure 120 on page 169 is displayed. Select **OSR3**. Click on it with the right-hand mouse button. On the pop-up menu, select the **recover** option.

  The OSR3-Recover window is displayed (see Figure 130).



Number of buffers: 2

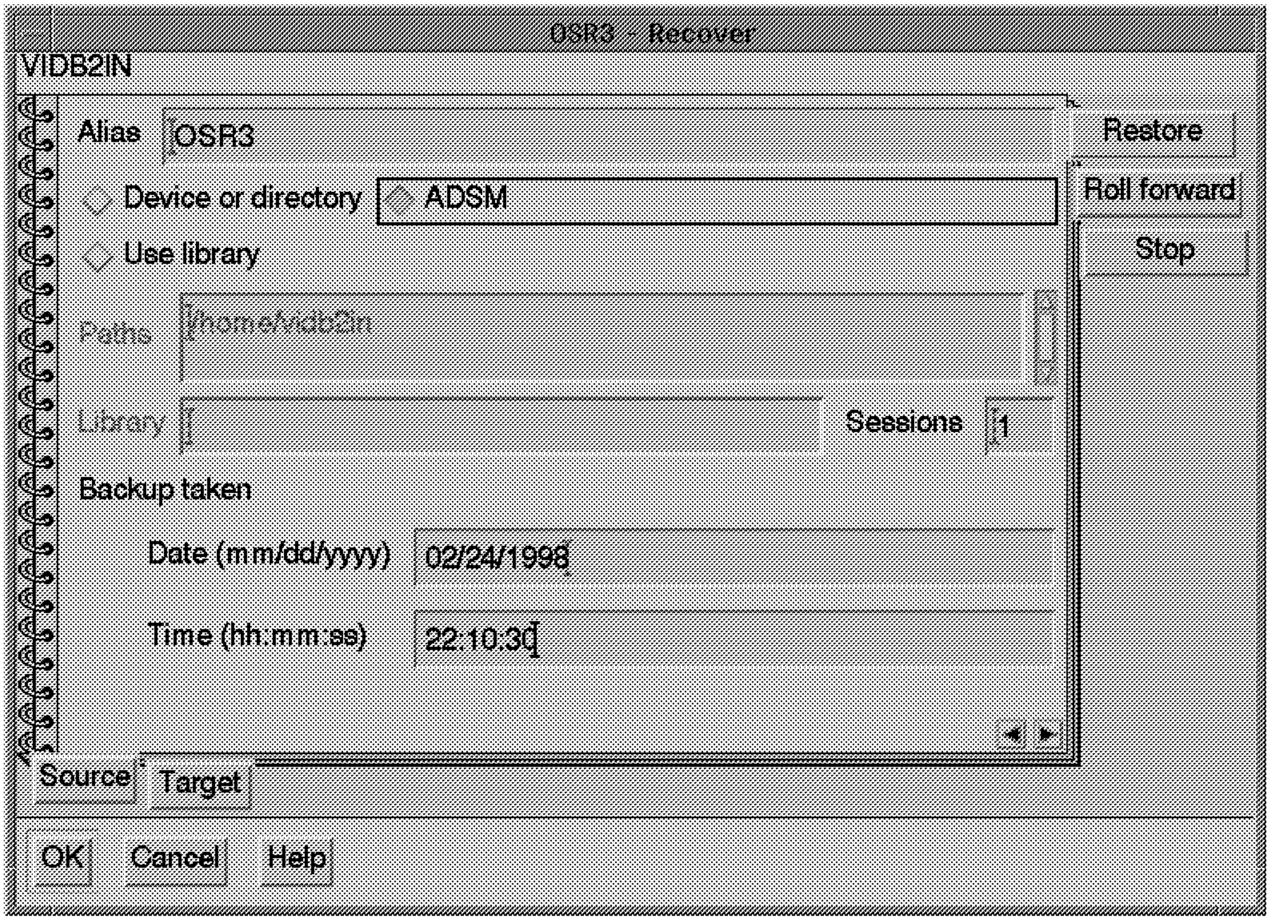
Size of each buffer (4KB):

Figure 130. OSR3 - Recover Window Recover Menu

Select the **Restore** option, then click on the right black arrow located at the bottom of this screen.

A new OSR3 Recover window is displayed. Select the following parameters as shown in Figure 131 on page 180.

- **Alias --> OSR3** (This is the alias of the database to recover.)
- **ADSM** (Use ADSM for recovering the database.)
- **Date (mm/dd/yyyy) --> 02/24/1998** (This is the start date of the database backup. This information is retrieved from the database backup time stamp.)
- **Time (hh:mm:ss) --> 22:10:30** (This is the start time of the database backup. This information is retrieved from the database backup time stamp.)



Alias OSR3

◇ Device or directory ◇ ADSM

Paths /home/vidb2in

Library

Sessions 1

Date (mm/dd/yyyy) 02/24/1998

Time (hh:mm:ss) 22:10:30

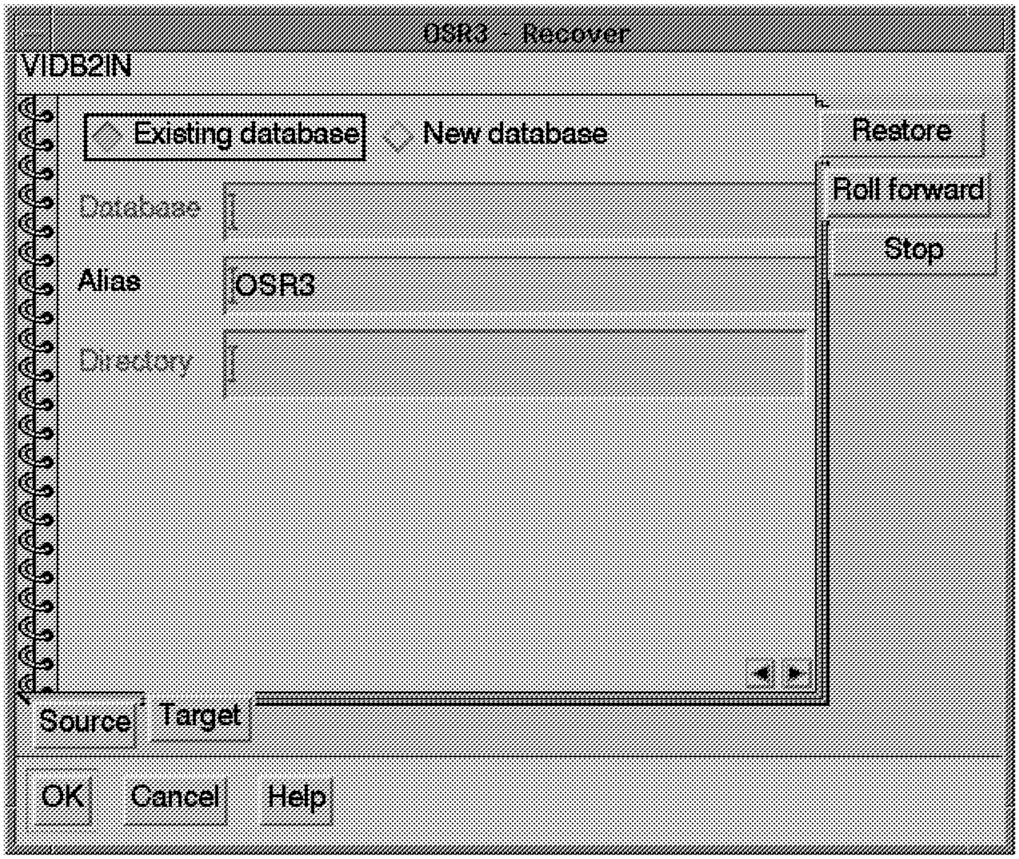
Figure 131. OSR3 - Recover Window Recover Menu

Then click on **Target**.

The OSR3-Recover Target window is displayed. Select the following parameters as shown in Figure 132 on page 181.

– **Existing database**

– **Alias --> OSR3**



Alias: OSR3

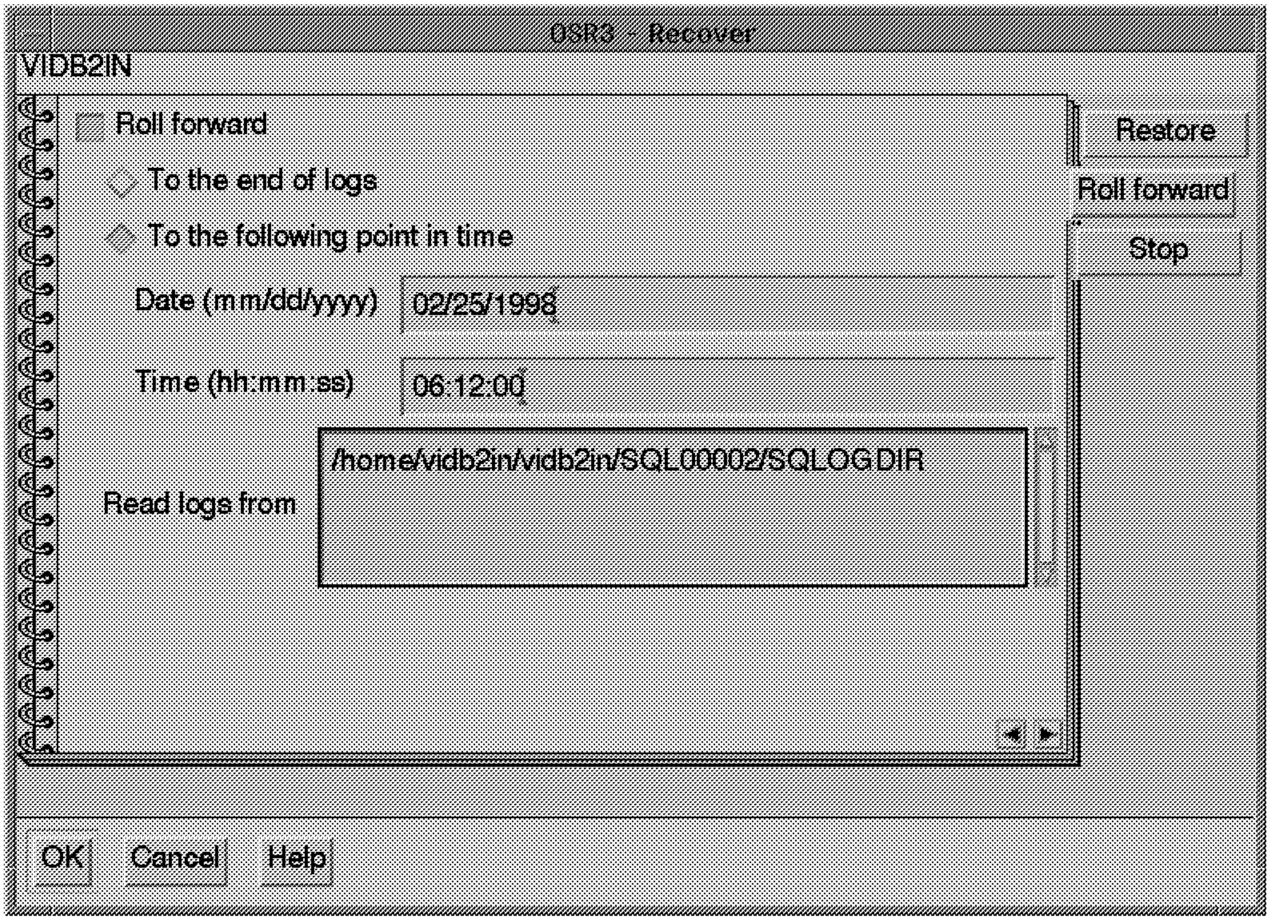
Directory:

Figure 132. OSR3 - Recover Window Target Menu

Then click on **Roll forward**.

The OSR3-Recover Roll forward window is displayed. Select the following parameters as shown in Figure 133 on page 182.

- **Roll forward**
- **To the following point in time**
- **Date (mm/dd/yyyy) --> 02/25/1998** (This is the date until which the log should be used.)
- **Time (hh:mm:ss) --> 06:12:00** (This is the time until which the log should be used.)
- **Read logs from --> /home/vidb2in/vidb2in/SQL00002/SQLOGDIR** (This is the directory where the logs are retained.)



VIDB2IN

□ Roll forward

◇ To the end of logs

◆ To the following point in time

Date (mm/dd/yyyy) 02/25/1998

Time (hh:mm:ss) 06:12:00

Read logs from /home/vidb2in/vidb2in/SQL00002/SQLOGDIR

Figure 133. OSR3 - Recover Roll Forward Menu

Attention

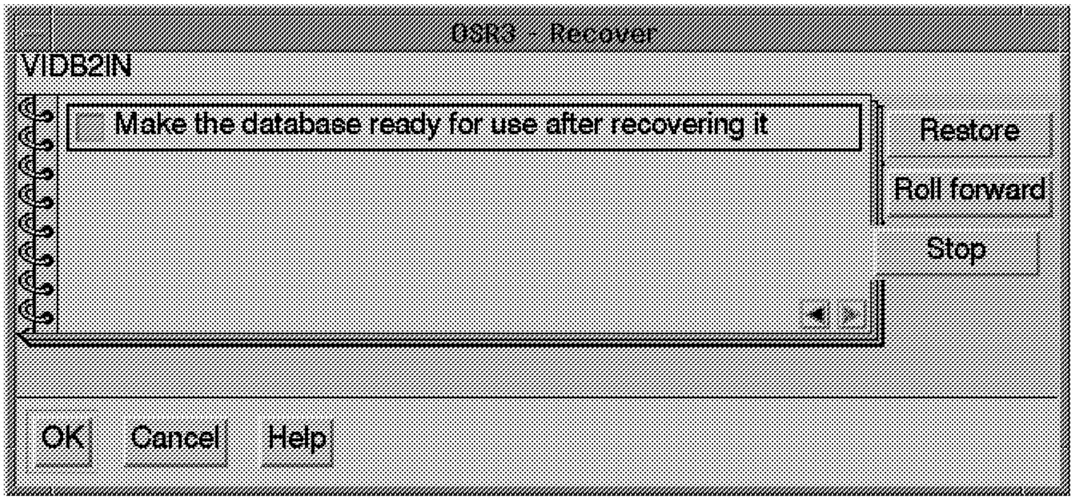
In order to restore a database without applying any transactions stored in the log files, roll forward is still mandatory with online backup.

In these date and time fields, you must specify a date and time greater than those of the online backup end, and they must be in Coordinated Universal Time (CUT) format. From San Jose, for example, it is eight hours ahead.

Then click on **Stop**.

The OSR3-Recover Stop window is displayed. Select the following parameters as shown in Figure 134 on page 183.

– **Make the database ready for use after recovering it**


OSR3 - Recover

VIDB2IN

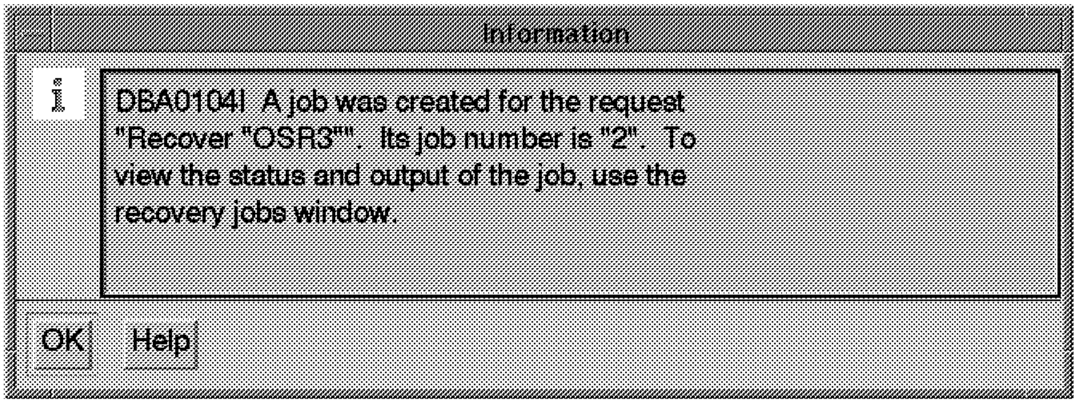
□ Make the database ready for use after recovering it

Figure 134. OSR3 - Recover Window Stop Menu

- Start the restore operation.

  Then click on **OK** to start the restoration of OSR3 database.

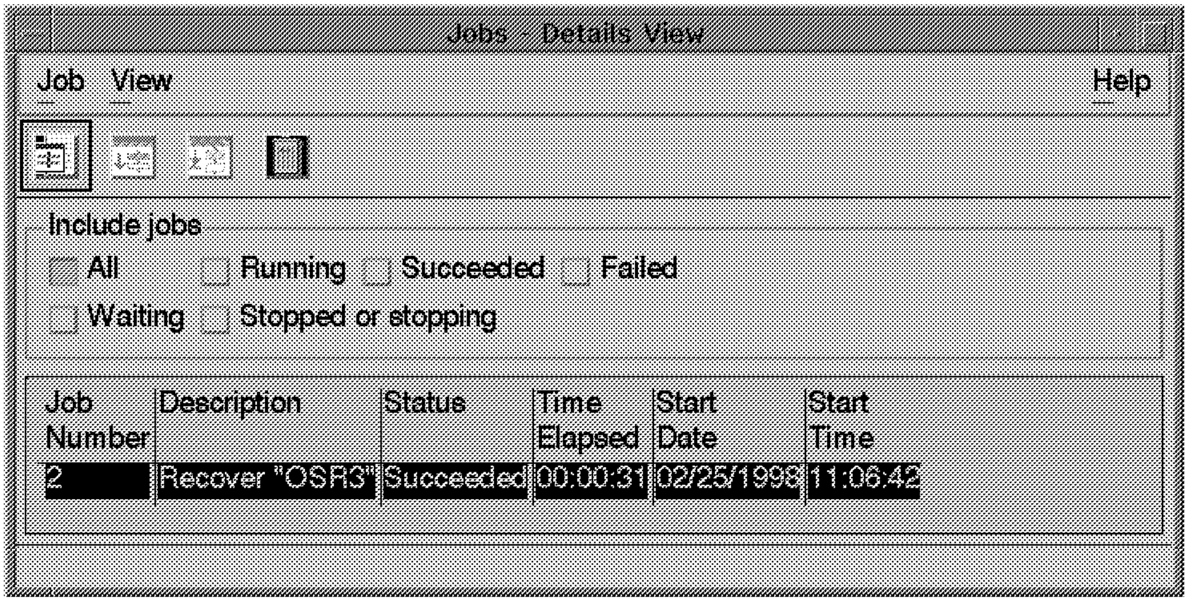
  An information window is displayed (see Figure 135).


Information

DBA0104I A job was created for the request "Recover "OSR3"". Its job number is "2". To view the status and output of the job, use the recovery jobs window.

Figure 135. Restore Job Number Information

To verify the status of this restore operation, return to the Database Director window (Figure 120 on page 169) and select **Windows** from the menu bar. On the pop-up, select **Recovery Status** to get the Jobs Details View window (Figure 136 on page 184).



Include jobs

■ All ☐ Running ☐ Succeeded ☐ Failed

☐ Waiting ☐ Stopped or stopping

Job Number	Description	Status	Time Elapsed	Start Date	Start Time
2	Recover "OSR3"	Succeeded	00:00:31	02/25/1998	11:06:42

Figure 136. Restore Job Status

Verify that the status of Job Number 2 is set to Succeeded. If so, the OSR3 database is successfully recovered.

2. Library server LSR3 database recovery

   - Stop the library server application by using the viadmin user ID.
   - Use the previous restore procedure to configure and run LSR3 recovery.

     At the end of the procedure, the LSR3 database is recovered.

Important

When making database backups each night, as in our example, it is necessary to manage both of the following:

- The ADSM volumes in which these image databases are stored
- The directory in which the DB2/6000 retain log are stored

Otherwise, these volumes and directory can run out of space and lead to a database schedule operation failure. For this purpose, DB2/6000 provides the db2adutl utility which allows you to extract, query and delete backup images, logs and load copy images saved using ADSM.

To be efficient for the image databases stored on ADSM, the copy group defined on the default management class MC4BU used for backup files and databases must be updated; its parameter *retextra* should be set to 0. This parameter specifies how many days ADSM retains a backup version after that version becomes inactive.

3. Directory /LONGTERM/ubosdata recovery

   - Start the ADSM backup/restore application by executing the following command:

     ```
     # dsm
     ```

   - The ADSM View windows is displayed (see Figure 76 on page 106). Click on **Restore files and directories to your system**.

- The Restore window is displayed. Click on the **+** icon next to ROLM3. Its backup files tree is displayed. Click on the **+** icon next to /LONGTERM. Its subdirectories are displayed. Select the **ubosdata** directory as shown in Figure 137 on page 185.

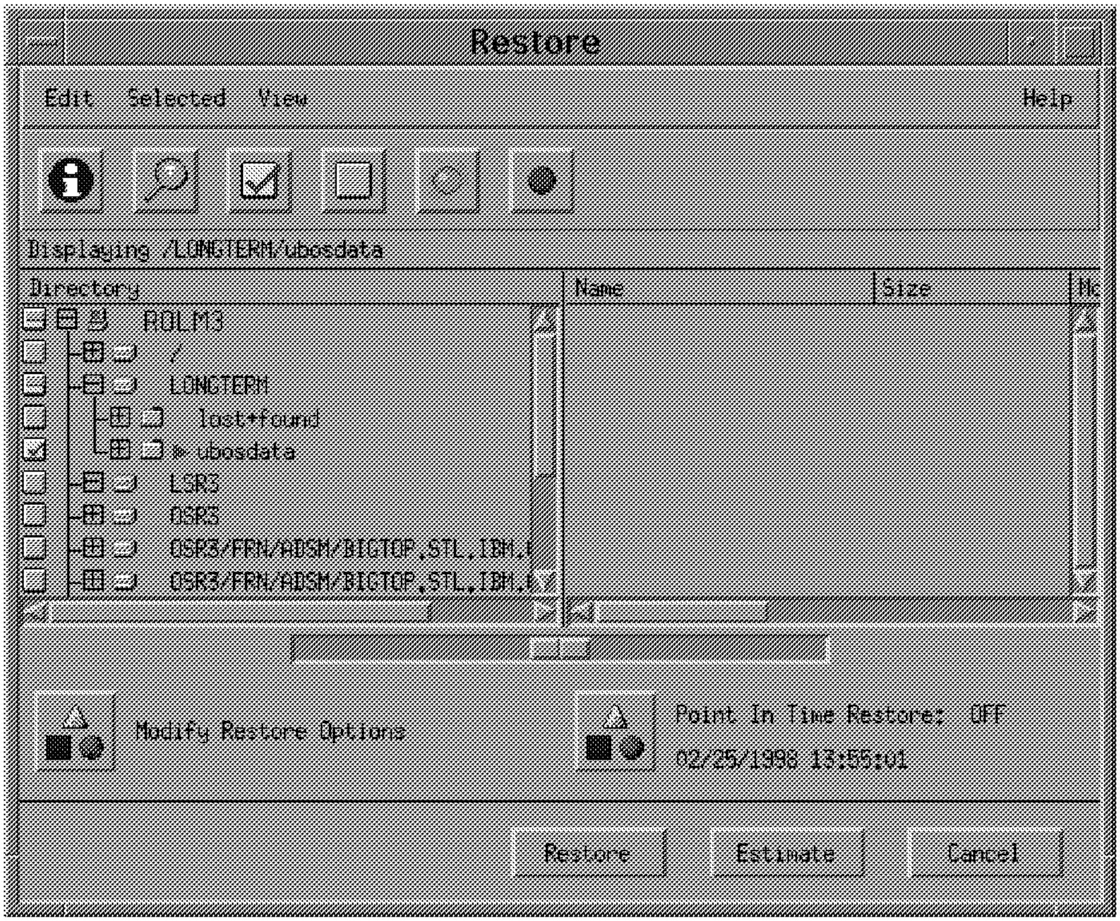
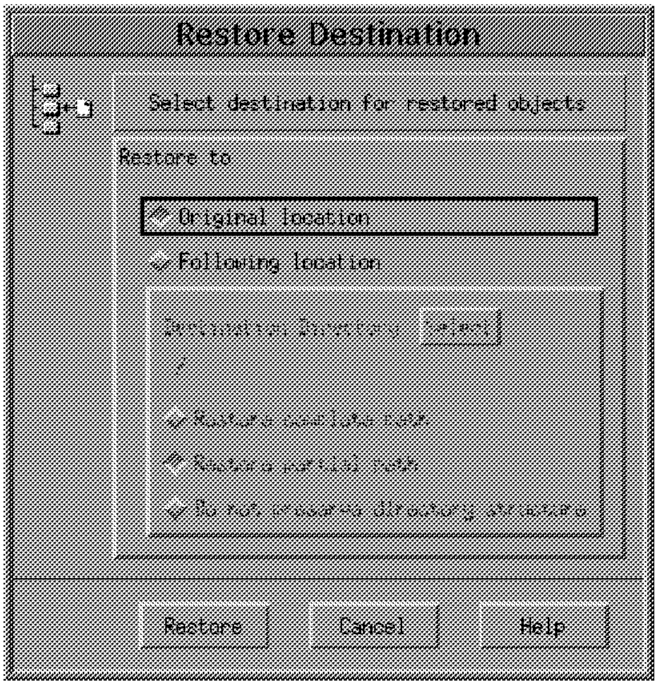


Figure 137. Selected Files to Restore

- Then click on **Restore**.

  The Restore Destination window is displayed. Select the parameters as shown in Figure 138 on page 186.

  – **Original location**



Restore Destination

Select destination for restored objects

Restore to

◆ Original location

◇ Following location

Destination Directory: Select

/

◇ Restore complete path

◆ Restore partial path

◇ Do not preserve directory structure

Figure 138. File Restore Destination

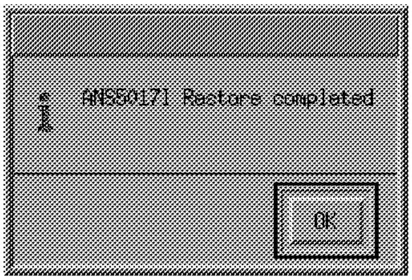
- Then click on **Restore** to start recovering files from the selected directory /LONGTERM/ubosdata.

  If the Files Exits windows is displayed, select the following option:

  – **Apply action to all remaining files**

  then click on **Replace**.

Once the recovery operation completes successfully, the pop-up menu (see Figure 139) is displayed.



ANS5017I Restore completed

Figure 139. Restore Result

At this point, the /LONGTERM/ubosdata directory is successfully recovered.

4. Directory /SHORTTERM/ubosdata recovery

   Run the same procedure to restore the directory /SHORTTERM/ubosdata.

   At the end of the procedure, the /SHORTTERM/ubosdata directory is successfully recovered.

5. Directory /home/viadmin2/ubosstg recovery

   Run the same procedure to restore the directory /home/viadmin2/ubosstg.

At the end of the procedure, the /home/viadmin2/ubosstg directory is successfully recovered.

Because no index classes have been created since last backup operation, it is not necessary to restore the /home/viadmin/frn/LSR3/dll directory.

Once all the recovery steps have completed successfully, we start the library server, then the object server. Then we open the VisualInfo client application and check that document retrieval works correctly.

Appendix A. Updating SMS Parameters through APIs

All you need to define SMS objects and configure your system is the System Administration program. But you can develop your own utilities, using the SMS APIs. One of those APIs, the SimLibChangeObjectSMS, is used in the NT example (#1) in this redbook. In this appendix we show you how to use the API and what you can do with it.

The format of the API is:

```
SimLibChangeObjectSMS
(hSession, hObj, pSMS, fChangeControl, pAsyncCtl, pRC)
```

All of the parameters are well known, except pSMS which is a pointer to a system-managed storage structure for an object. This structure consists of:

```
ULONG ulStruct;
LONG lSMSRetention;
CHAR szCollectionName;
CHAR szObjectName;
CHAR szMgtCls;
CHAR szStgCls;
CHAR szDataCls;
CHAR szStoreSite;
CHAR szStoreHint;
```

where:

- ulStruct is the length of the structure in bytes.
- lSMSRetention is the period in days that the system retains the object in system-managed storage. This field should be set to 0; it is not used in the current VisualInfo release.
- szCollectionName is the collection name. Remember that after an object has been assigned to a collection on an object server, you cannot change the collection assignment. Therefore this parameter cannot be changed, and you must use the value of the current collection of the object.
- szObjectName is a parameter not used in the current release, so you can use any value.
- szMgtCls is the name of the management class. You can use the name of the new management class you want to assign to the object.
- szStgCls is the name of the storage class. You can use the name of the new storage class you want to assign to the object.
- szDataCls is a parameter not used in the current release, so you can use any value.
- szStoreSite is the name of the object server. You cannot change this value with this function, so use the name of the object server where the object is stored.
- szStoreHint is a parameter not used in the current release, so you can use any value.

Guidelines for use:

- If you do not want to change a field, set it to NULL.

- If the SMS change causes any physical movement of the object, it might be deferred until the storage management cycle runs on the server.
- You cannot move the object from one server to another, using this function.

We can examine how to use this API in the NT example (Example 1) of this book. Remember that we have objects in collection CBR.CLLCTB, with management class MGTCLASB, and we want to change the management class to MGTCLASA. The steps you must follow are:

1. Assign the correct values to the SMS structure:

```
SMS SMSData ={
   sizeof(SMSData),
   0,
   ″CBR.CLLCTB″,
   ″IGNORE″
   ″MGTCLASA″,
   ″FIXED″,
   ″IGNORE″,
   ″OBJSRVRN″,
   ″IGNORE″,
   };
```

2. Execute the SimLibChangeObjectSMS API:

```
SimLibChangeObjectSMS(
   (HSESSION)hSession,
   (HOBJ)&Obj,
   (PSMS)&SMSData,
   (BITS)0,
   (PASYNCTLSTRUCT)NULL,
   (PRCSTRUCT)&RCStruct
   );
```

The object will stay on DASD for 10 days. Then it is migrated to optical disks, as specified in its new management class.

Appendix B. Troubleshooting

If you encounter problems, you might find the solutions in this appendix. Here we list the problems we ran into and describe how we fixed them.

B.1 Troubleshooting for Windows NT

In the following sections we discuss ADSM commands to use when troubleshooting for AIX, as well as some log and trace files to use. We give some tips for modifying ADSM and some performance hints for the object server.

B.1.1 ADSM Commands

You can use the QUERY command (or just the abbreviation q) to get detailed status of the parameters of your system. The following examples relate to the VisualInfo system:

- The first time the object server stores an object in ADSM, a /OBJSRVRN/FRN/ADSM//STANDARD filesystem is created, where OBJSRVRN is the name of your object server, and STANDARD is the name of the ADSM volume you created with the VisualInfo system administration program. This example gives information about the filespace used by the PEPE200 object server:

```
adsm> q filespace pepe200

Node Name                      Filespace   Platform Filespace Capacity   Pct
                               Name                 Type          (MB)  Util
------------------------------ ----------- -------- --------- -------- -----
PEPE200                        /OBJSRVRN/  FRNNADDM API:Digital    0.0   0.0
                                FRN/ADSM/           Library
                                /STANDARD           - API
```

- To determine the number of files and the space occupied in the optical library use this command:

```
adsm> q occupancy pepe200

Node Name         Type  Filespace   Storage     Number of  Physical     Logical
                        Name        Pool Name       Files     Space       Space
                                                           Occupied   Occupied
                                                               (MB)        (MB)
----------------  ----  ----------  ----------  ---------  ---------  ---------
PEPE200           Bkup  /OBJSRVRN/  C64OPTPOOL         50       4.67       4.67
                         FRN/ADSM/
                         /STANDARD
```

- To determine which volumes are in the library, use this command

```
adsm> q libvolume

Library Name     Volume Name     Status                Last Use
------------     -----------     ------------------    ---------
C64LIB           SVT001          Private(Optical)      Data
C64LIB           SVT002          Scratch(Optical)
C64LIB           SVT003          Scratch(Optical)
```

Note that volume SVT001 has a status of Private because VisualInfo uses it to store data.

- You can get information about each volume in standard format:

```
adsm> q volume svt001

Volume Name               Storage      Device       Estimated    Pct   Volume
                          Pool Name    Class Name   Capacity    Util   Status
                                                        (MB)
------------------------  -----------  ----------  ---------  -----  --------
SVT001                    C64OPTPOOL   C64OPT        2,486.3    0.6  Filling
```

or in detailed format:

```
adsm> q volume svt001 format=detail

                   Volume Name: SVT001
             Storage Pool Name: C64OPTPOOL
             Device Class Name: C64OPT
       Estimated Capacity (MB): 2,486.3
                      Pct Util: 0.6
                 Volume Status: Filling
                        Access: Read/Write
        Pct. Reclaimable Space: 0.9
               Scratch Volume?: Yes
               In Error State?: No
      Number of Writable Sides: 2
       Number of Times Mounted: 144
             Write Pass Number: 1
     Approx. Date Last Written: 02/03/1998 16:07:43
        Approx. Date Last Read: 01/30/1998 18:56:20
           Date Became Pending:
        Number of Write Errors: 0
         Number of Read Errors: 0
               Volume Location:
Last Update by (administrator):
         Last Update Date/Time: 01/22/1998 12:39:52
```

B.1.2 Log Files

You can refer to the following log files to determine where errors occurred.

B.1.2.1 VisualInfo

The VisualInfo log file is located in the FRNROOTLOG directory, and its name is FRNDIAG.LOG. All VisualInfo library and object server operations are recorded in this file. Use this file when you have a problem with your VisualInfo system.

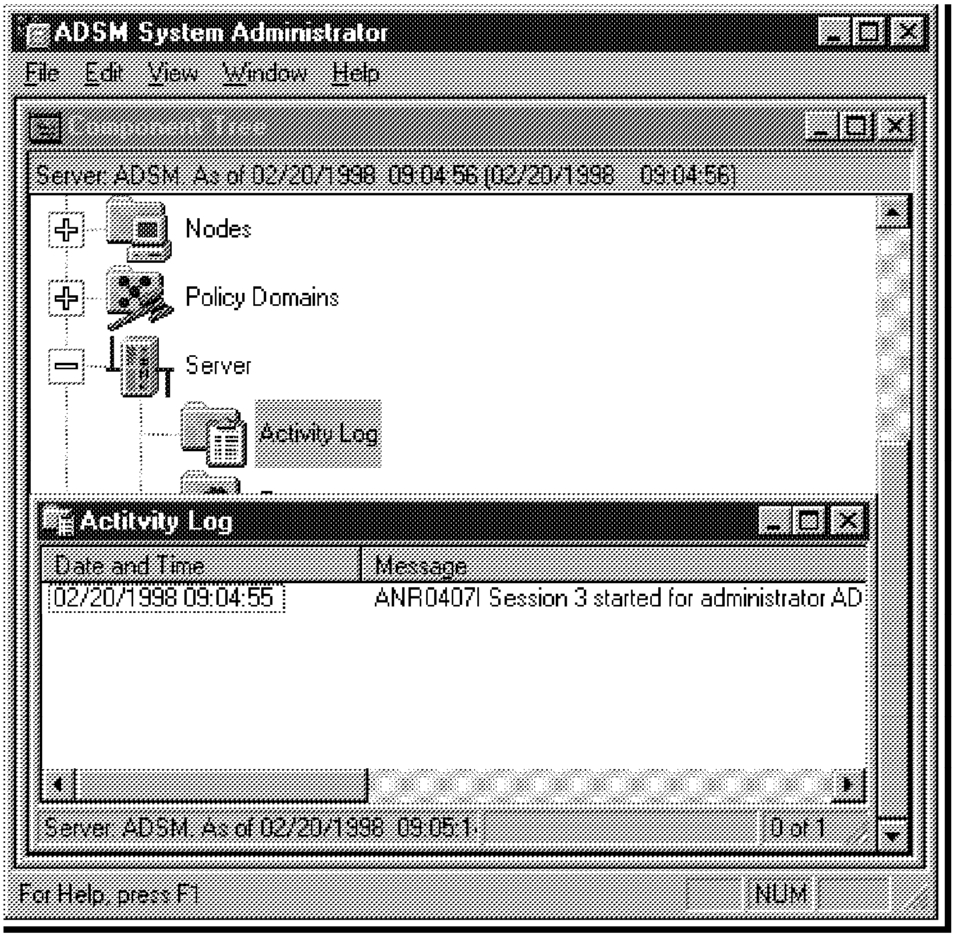
For example, what if the object server ends a few seconds after you start it, and there is no message? This may because the object server found some error while reading its tables or when communicating with ADSM. Examine the FRNDIAG.LOG to find the source of the error.

B.1.2.2 ADSM Client

The ADSM client log file is in FRNROOT\LOG directory and its name is DSIERROR.LOG. ADSM uses this file to record the operations made by the ADSM client used by the object server.

B.1.2.3 ADSM Activity

The ADSM activity log file is managed by the ADSM server. You can view it when you start an ADSM Administration client session: expand the tree under **Server** and double-click on **Activity log** (see Figure 140).



Server: ADSM. As of 02/20/1998 09:04:56 [02/20/1998 09:04:56]

Date and Time	Message
02/20/1998 09:04:55	ANR0407I Session 3 started for administrator AD

Server: ADSM. As of 02/20/1998 09:05:1 0 of 1

Figure 140. ADSM Activity Log

B.1.2.4 Windows NT Application Log

This application log is in the standard NT event log. You can view it by selecting **Start, Programs, Administration Tools, Event Viewer** in the NT desktop. Error messages generated by VisualInfo components are identified by FRN in the Source column.

B.1.3 Traces

Traces are powerful diagnostic utilities but frequently are difficult to understand. You will need to refer to them when you have a severe error or if you want to learn more about the functioning of the system. For a detailed trace of the communication between the object server and ADSM, insert these two lines in the dsm.opt file of the c:\win32app\ibm\adsm\api\dll directory:

```
TRACEFL INSTR_CLIENT_DETAIL FS API PID COMM SESSION
TRACEFILE C:\FRNROOT\LOG\ADSM.TRC
```

For an explanation of those parameters, see the *ADSM Trace Facility Guide*.

You can get traces from VisualInfo servers, but they are most useful when you are in contact with the support center. See *ImagePlus VisualInfo: Messages and Codes*, SC31-9065 (softcopy only) for a description of those traces.

B.1.4 Modifying the Default ADSM Parameters

The object server writes to ADSM using an ADSM backup copy group. Because there is no need for multiple generations of the object server′s objects, you can set the maximum number of backup copies to 1. You can choose to maintain more than one copy, but the VisualInfo object server does not acknowledge use of more than one backup copy.

You should also increase the ADSM server session timeout parameter (IDLETIMEOUT in the dsmserv.opt file) if you want to maintain longer service connections between the object server and the ADSM server. Increasing the timeout parameters reduces the overall overhead if sessions need to be re-established regularly.

B.1.5 Object Server Performance Tips

Here are some performance tips to consider when implementing your NT system:

- Do not migrate objects during periods of high activity.
- Avoid filling the staging area.
- An ideal environment is one where destaging never occurs (the Staging Percent Start is never reached) during prime working hours. Save it for off hours, if possible. Therefore your staging area should be large enough to hold all objects to be stored during a single day plus objects that are retrieved during the day. If this is not possible, attempt to maintain an object store rate that does not overrun the destager.

  An alternative to avoid destaging during the day is to store objects after the work day completes and destage and migrate during off hours. This approach prepares your work for the following day.
- Isolate the staging area in a separate physical drive.
- Distribute disk I/O by keeping the DB2 log files and the server database on separate physical drives.
- Reorganize the BASE_OBJECTS table frequently, if there are many object store requests.
- Destaging or migrating to ADSM-managed optical disks is significantly slower than destaging or migrating to ADSM-managed DASD. Therefore, it is recommended that you destage or migrate to ADSM-managed DASD first, and let ADSM handle the migration between unlike I/O devices, such as DASD, optical, and tape. If you use ADSM for migration, the object server migrator will not have to run at all, or less often, leaving more CPU and I/O cycles for other object server activities.
- Optical retrievals are significantly slower than DASD retrievals. Therefore, for better performance, keep frequently accessed objects on DASD and use optical device to store objects that are rarely accessed.

B.2 Troubleshooting for AIX

In the following sections we discuss ADSM commands to use when troubleshooting for AIX, as well as common problems encountered when setting up Example 2. We give some tips related to the use of the subsystems we installed and configured.

B.2.1 ADSM Commands

Following are ADSM administrator′s commands to use when working with volumes:

1. Status of mounted volume

   To know which volume is mounted in one of the two drives and its related status, execute the following command:

   ```
   adsm> query mount
   ```

   In our example, the output is:

   ```
   ANR8330I OPTICAL volume VI001A is mounted R/W in drive DRV1 (/dev/rop1)
            status: IN USE.
   ```

2. Status of a volume

   To display information about a volume, execute the following command: (volume VI001A in our example)

   ```
   adsm> query volume VI001A format=detailed
   ```

   In our example the output is:

   ```
   Volume Name: VI001A
   Storage Pool Name: SPE2ST
   Device Class Name: DCOPTI
   Estimated Capacity (MB): 2,486.3
    Pct Util: 0.6
   Volume Status: Filling
   Access: Read/Write
   Pct. Reclaimable Space: 0.0
   Scratch Volume?: No
   In Error State?: No
   Number of Writable Sides: 2
   Number of Times Mounted: 2
   Write Pass Number: 1
   Approx. Date Last Written: 02/02/98 09:50:01
   Approx. Date Last Read: 01/30/98 22:28:53
   Date Became Pending:
   Number of Write Errors: 0
   Number of Read Errors:  0
   Volume Location:
   Last Update by (administrator): ADMIN
   Last Update Date/Time: 01/30/98 22:09:42
   ```

   If you have write errors on a volume, ADSM changes the volume access status from Read/Write to ReadOnly. Once I/O errors are corrected, modify the volume access status in order to use it again. The following command should be executed:

   ```
   adsm> update volume VI001A access=readwrite
   ```

3. Dismount a volume

If you need to check out a volume inserted in a drive, the checkout command fails because the drive is in use. Dismount the volume first. Use this command to dismount volume VI001A:

```
adsm> dismount volume VI001A
```

B.2.2 VisualInfo Installation Problems

The following are common problems encountered during the installation of VisualInfo and the solutions to those problems.

- The TaskGuide for the VisualInfo Installer, after starting the few text lines, says `unable to open display`.

  To solve this problem, make sure that you have a VGA monitor and supporting display driver. Also enter the exact hostname in the DISPLAY environment variable, as

  ```
  export DISPLAY=hostname:0
  ```

  where hostname is the workstation hostname where you are installing VisualInfo.

- If you have not installed JDK 1.1.2, you will not be able to install VisualInfo 2.3.1 for AIX. TaskGuide needs a Java run-time environment.

  To solve this problem, ensure that JDK 1.1.2 is installed.

- If you follow the installation instructions and prerequisites, you should not have any other problems. If you do, exit TaskGuide and look for the frnxinst.log file in the directory /tmp/frn for error messages.

B.2.3 VisualInfo Configuration Problems

You could have problems with AIX or VisualInfo configurations. Make sure that the users and file systems (which VisualInfo uses for defining volumes) are created by an AIX expert. We discuss some typical configuration problems and their solutions.

- When you try to log on to the library server that is listed in the server listbox, the client application either hangs or returns the message “Server not responding.”

  To solve this problem, before logging onto the library server, verify the network connections for both the object server and the library server, using the utility provided with the client application. If the network connection verification fails, make sure that both library server and object server are running; also check whether TCP/IP is working by pinging from both ends. Ensure also that Winsocks is working on the client workstation.

  If the network connection verification does not report an error but hangs instead, make sure that you supplied the proper server name, port number, communication protocol, hostname of the server, and operating system of the server. If any one of these differ from the configured variables during network table generation, the system will hang.

  In this case, kill the client application, stop the servers, and run `frnkill.all` in each session of the servers. Then make sure that process `frnliis` is not running. If it is, kill that process also from an AIX prompt.

  Regenerate the network table by giving proper network parameters, which were assigned during installation.

- From VisualInfo system administration, when you try to define object server SMS components, you get an error saying the LAN-based object server is still active.

  To solve this problem, make sure that your object server is down and your SMS server is up, along with your library server. Here you need only a library server and SMS server while defining SMS components.

- When you do **File-->Import** from client application, the client hangs.

  To solve this problem, make sure you have a C Set ++ license, and you have executed db2ln.

  1. For the C Set ++ license, verify as described in the installation section. If you need to install it, contact the appropriate support person. Also check the directory */home/viadmin/frn/Lib_SrvDBname/dll*. If it is empty, you need to recreate the library server database, after ensuring that you have a C Set ++ license and the following files:

     – /home/netls/glb_obj.txt

     – /home/netls/glb_site.txt

  2. For DB2, check the directory */usr/bin* for links to the DB2 bin directory where installed. If there are no links to DB2 files, run db2ln to create them.

     Also check the directory /home/Lib_Srv_userid/frn/Lib_Srv_dbname/dll for the existence of a set of DLLs from when the database was created. If empty, run the db2ln command, then check that directory again for DLLs.

B.2.4 VisualInfo Startup Problems

Sometimes when you try to bring up the library server, the object server, or the SMS server, one of these may not start. In this section we discuss such problems and their solutions.

1. When you try to start an object server, it may come up but go down after initializing.

   This happens if you stopped the object server improperly, or killed the object server process. To solve this problem, recover the object server database by issuing the frnrecdb.obj command, then restart the object server. The error log file for the object server is /home/Obj_Srv_userid/frn/log/frndiag.log, which you can look at for further error information.

2. An object server fails to recover its object server database.

   If you did not run frnkill.all before recovering a database, you would see this. To solve this problem, ensure that you stop all VisualInfo processes including the library server, object server, and SMS server, to clean up system resources. Execute *frnkill.all* for each process.

3. When you select **Bring up SMS server**, the server closes down.

   If you stopped the library server or SMS server improperly, you face this. To solve this problem, stop the library server, then run *frnkill.all*. In the SMS session, also run *frnkill.all* to clean up the running and hung processes. Now start the library server and SMS server with their user IDs. If the same problem persists, kill all frn processes from AIX command. Start the servers again.

You can find the errors for these problems under the directory /home/SMS_user_id/frn/log in the file frndiag.log.

B.2.5 VisualInfo Operational Problems

The following problems arise if steps or procedures are not followed correctly. If you experience one of these problems, ensure that you have followed the procedure for that process. Here is an example of a problems which you could encounter:

1. You cannot view objects of GIF format from the VisualInfo client application.

To solve problems such as this, look in the VisualInfo error logs. The VisualInfo system generates error log files whenever there is a problem which needs attention of the specialist.

Here are some log files to look in during operational problems:

- Library server: /home/Lib_Srv_userid/frn/log/frndiag.log
- Object server: /home/Obj_Srv_userid/frn/log/frndiag.log
- SMS server: /home/SMS_Srv_userid/frn/log/frndiag.log

B.2.6 AIX-Related Problems

Sometimes when certain file permissions and user rights are not assigned properly, your VisualInfo system might hang or give an error. Or it might report an error, but the result will not be seen in files where they should reside. Here is an example:

> You define a VisualInfo SMS DASD volume in AIX and define a migration policy for the objects. After doing this, you will not find objects in the volume which is in the AIX filesystem.
>
> The problem is that when you created a file system to be used for defining VisualInfo DASD volume as root user, and did not ensure permission for VisualInfo servers′ user IDs, then the object server user ID will not have permission to write into that filesystem.
>
> To solve this problem, ensure that the *rwx* permissions apply to VisualInfo servers user IDs.

B.2.7 Object Server Performance Tips

Following are some performance tips you can consider when implementing your AIX system:

- Avoid migrating objects during periods of high online activity.
- Avoid filling the staging area. When the object server cannot store an object, it initiates the purger process. If the purger cannot clear enough space, the destager process is started. The purger/destager take cycles away from the online activities.
- Ideal, the staging area should be large enough to hold objects to be stored and retrieved during the online day. This would mean that destaging occurs during the off hour.
- Another way to avoid destaging during the online day is to store objects after the work day completes.

- Isolate the staging area in a separate physical drive
- Keep the DB2/6000 log files and the DB2/6000 object server database on separate physical drives.
- Reorganize the BASE_OBJECTS table frequently.

Appendix C. ADSM APIs Used by VisualInfo

Following are the ADSM APIs used by VisualInfo.

- dsmInit

  This API is used to start an ADSM API session and connect the client to ADSM storage. The application client can only have one active session at a time. To open another session with different parameters, the dsm Terminate call must first be used to end the current session.

- dsmQUerySession

  This API is used to initiate a query request to ADSM for all information related to the operation of the specified session in *dsmHandle*.

- dsmBeginQuery

  This API is used to start a restore/retrieve operation on a list of objects in ADSM storage.

- dsmGetNextQObj

  This API is used to get the next query response from a previous dsmBeginQuery call and place it in the caller′s buffer.

- dsmEndQuery

  This API is used to signify the end of a dsmBeginQuery action.

- dsmTerminate

  This API is used to end a session with the ADSM server and clean up the ADSM environment.

- dsmBeginTxn

  This API is used to begin an ADSM transaction. This call indicates to ADSM the beginning of one or more actions that are executed as an automatic unit; that is, all the actions succeed or none succeed.

- dsmBindMC

  This API is used to associate (or bind) an ADSM management class to the passed object.

- dsmSendObj

  This API is used to initiate a request to send a single object to ADSM storage.

- dsmSendData

  This API is used to send a byte stream of data to ADSM via buffer.

- dsmEndSendObj

  This API is used to signify the end of data being sent to ADSM storage.

- dsmEndTxn

  This API is used to end an ADSM transaction.

- dsmUpdateFS

  This API is used to update a file space in ADSM storage. This ensures that the ADSM administrator has an up-to-date record of your file space.

Appendix D. Special Notices

This publication is intended to help technical specialists and consultants to configure the System Managed Storage component of ImagePlus VisualInfo or Digital Library on the Windows NT or AIX platforms. The information in this publication is not intended as the specification of any programming interfaces that are provided by VisualInfo or Digital Library. See the PUBLICATIONS section of the IBM Programming Announcement for VisualInfo or Digital Library for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

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:

AD/Cycle®
ADSTAR®
Advanced Peer-to-Peer Networking®
AIX®
APPN®
AS/400®
AT®
BookManager®
CICS®
Common User Access®
CUA®
DATABASE 2
DB2®
DRDA®
FaxConcentrator®
FFST
FlowMark®
IBM®
ImagePlus®
MO:DCA
MVS® (logo)
MVS/ESA
OS/2®
OS/400®
Presentation Manager®
PS/2®
QBIC
RS/6000
S/390®
SAA®
SOM (NOT FOR USE IN JAPAN)
System/390®
Systems Application Architecture®
SystemView®
VisualAge®
VisualInfo
VTAM®
XGA®
400®

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 E. Related Publications

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

E.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 207.

- VisualInfo or Digital Library
  - *Best Practices for VisualInfo*, SG24-4792
  - *Getting Started with Digital Library V1.0*, SG24-2243
  - *Analyzing VisualInfo Performance and Capacity*, SG24-4974
  - *A Simple Approach to VisualInfo*, GG24-4444
  - *Imageplus VisualInfo Client/Server Solution: Sample Code for Client*, GG24-4369
  - *ImagePlus VisualInfo Library and Object Servers for MVS/ESA Planning Guide*, GG24-4452
  - *ImagePlus VisualInfo MVS/ESA Installation Guide*, GG24-4488
- ADSM
  - *ADSM Server for Windows NT Configuration and Recovery Examples*, SG24-4878
  - *ADSM API Examples for OS/2 and Windows*, SG24-2588
  - *Using ADSM to Back Up Databases*, SG24-4335
  - *ADSM Concepts*, SG24-4877
  - *ADSM Version 3 Technical Guide*, SG24-2236
  - *ADSM Operations and Management with TME 10*, SG24-2214

E.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

E.3 Other Publications

These publications are also relevant as further information sources:

- VisualInfo or Digital Library
  - *ImagePlus VisualInfo: Getting Started*, GC31-9051
  - *ImagePlus VisualInfo: Planning and Installation Guide*, GC31-7772
  - *ImagePlus VisualInfo: System Administration Guide*, SC31-7774
  - *ImagePlus VisualInfo: Client for OS/2 User′s Guide*, SC31-7775
  - *ImagePlus VisualInfo: Client for Windows User′s Guide*, SC31-9052
- ADSM
  - *ADSM V3R1 Windows NT Admin Guide*, GC35-0292
  - *ADSM V3R1 Windows NT Reference*, GC35-0293
  - *ADSM V3R1 Windows NT Quick Start*, GC35-0295
  - *ADSM V3R1 AIX Quick Start*, GC35-0273
  - *ADSM V3R1 AIX Admin Guide*, GC35-0274
  - *ADSM V3R1 AIX Reference*, GC35-0275

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 Allerobject serverd
Denmark

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (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.

Glossary

This glossary defines all terms and abbreviations used in this publication that may be new or unfamiliar as well as previously defined terms that you may want to look up.

This glossary includes terms and definitions from:

- The *IBM Dictionary of Computing*, SC20-1699.
- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- The *ANSI/EIA Standard - 440A: Fiber Optic Terminology*, copyright 1989 by the Electronics Industries Association (EIA). Copies can be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue N.W., Washington, D.C. 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

Deprecated term for. This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the dictionary.

See. This refers the reader to multiple-word terms in which this term appears.

See also. This refers the reader to terms that have a related, but not synonymous, meaning.

Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.

A

access. (1) To obtain the use of a computer resource. (T) (2) To obtain data from or to put data in storage. (3) In computer security, a specific type of interaction between a subject and an object that results in the flow of information from one to the other. See *read access*, *write access*.

access control. (1) In computer security, a method of ensuring that only authorized users can access the resources of a computer in authorized ways. (2) In the Digital Library system, access control provides the capability to limit access to certain functions provided by the system and certain objects stored in the system. You use the system administration program to create access lists to manage access control.

access level. The level of authority a subject has when using a protected resource; for example, authority to access a particular security level of information.

access list. (1) A collection of access rights for various users or groups for one object. (2) In computer security, a list associated with an object that identifies all the subjects that can access the object and their access rights; for example, a list associated with a file that identifies users who can access the file and identifies their access rights to that file. (3) A list consisting of one or more individual user IDs and user groups and the privilege set associated with each user ID or user group.

You can use an access list to limit access to an index class by associating the access list with that index class. This association further defines the privileges specified in the privilege sets previously associated with the individual user IDs or with the user IDs defined in the user groups specified in the access list.

You use the system administration program to create access lists to manage access control.

access path. (1) The sequence of activities required to locate, retrieve, or store data. (2) In a database, the procedure used by a database management system to access data.

access right. (1) Permission for a subject to use an access type for a particular object; for example, permission for a process to read a file. (2) The right to use a defined computer resource such as a library or file.

action. One of the defined tasks that an application performs. Actions modify the properties of an object or manipulate the object in some way.

action code. A number or letter assigned to an action in an action list.

action message. A message that indicates that a condition has occurred that requires a response from the user. The user can correct the condition and then continue using the product, withdraw the request for action, or get help. See also *information message*, *warning message*.

address. (1) A character or group of characters that identifies a register, a particular part of storage, or some other source or destination. (A) (2) The location in the storage of a computer where data is stored. (3) A name, label, or number identifying a location in storage, a device in a system or network, or any other data source. (4) In data communication, the unique code assigned to each device or computer connected to a network.

administrator. See *system administrator*.

advanced peer-to-peer networking (APPN). Data communication support that routes data in a network between two or more APPC systems that do not need to be adjacent.

after-image. A copy of a block of a record after a modification. (T) Contrast with *before-image*.

AIX. IBM′s implementation of the UNIX** operating system that features multiprogramming in a multi-user environment.

algorithm. An ordered set of well-defined rules for the solution of a problem in a finite number of steps.

alias. (1) An alternative label; for example, a label and one or more aliases may be used to refer to the same data element or point in a computer program. (A) (2) An alternative name for a member of a partitioned data set.

align. To arrange in a column in which all values either start in the same position (left-aligned) or end in the same position (right-aligned). ***Note:*** This operation can differ for bidirectional and double-byte character set (DBCS) languages.

alphabetic. Pertaining to data that consists of letters and other symbols used in language but excludes digits. Contrast with *alphanumeric, numeric*.

alphanumeric. Pertaining to data that consists of letters, digits, and usually other characters, such as punctuation marks. (T) Contrast with *numeric*, *alphabetic*.

API. Application programming interface.

append. (1) A function or mode that enables a user to add a new character string to the end of previously entered text. (T) (2) In the ImagePlus system, a function or mode that enables a user to add new text to the end of an existing note.

application. (1) The use to which an information processing system is put; for example, a payroll application, an airline reservation application, a network application. (2) A collection of related components with which a user performs a task.

application program. (1) A program that is specific to the solution of an application problem. Synonymous with application software. (T) (2) A program written for or by a user that applies to the user′s work, such as a program that does inventory control or payroll, or the Client Application provided with the Digital Library system.

application programming interface (API). The formally defined programming language interface between an IBM system control program or a licensed program and the user of the program.

APPN. Advanced peer-to-peer networking.

architecture. (1) The organizational structure of a system or component. See also *component*. (2) A blueprint for development that usually offers a standard set of services or common functions that can be used through a standard set of interfaces.

archive. (1) A copy of one or more files or a copy of a database that is saved for future reference or for recovery purposes in case the original data is damaged or lost. (2) To store backup copies of files for future reference or recovery purposes.

array. An arrangement of data in one or more dimensions: a list, a table, or a multidimensional arrangement of items.

ASCII. American National Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters (8-bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A) ***Note:*** IBM has defined an extension to ASCII code (characters 128-255).

ASYNC. Asynchronous.

asynchronous (ASYNC). Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T) Contrast with *synchronous*.

asynchronous processing. A series of operations performed separately from the job in which they were requested; for example, submitting a batch job from an interactive job at a computer.

attribute. (1) A named property of an entity. (2) A characteristic of an item.

attribute values table (AVT). A database table that is generated when an index class is defined. The table contains a column for each attribute of the index class. A row is added to the table each time an item is assigned to the index class.

authority. The right to access objects, resources, or functions.

authorization. (1) A user′s right to communicate with or make use of a computer system. (2) An access right. (3) The process of granting a user either complete or restricted access to an object, resource, or function.

availability. (1) The degree to which a system or component is operational and accessible when required for use. Often expressed as a probability. (2) In computer security, the property of being accessible and usable on demand by an authorized subject.

AVT. Attribute values table.

B

back up. To copy information, usually on a diskette, tape, or optical disk, for safekeeping.

backup. (1) Pertaining to a procedure, technique, or hardware used to recover lost or destroyed data or to keep a system operating. (T) (2) Pertaining to a system, device, file, or facility that can be used in the event of a malfunction or loss of data. See *back up*.

backup file. A copy of a file made for possible later reconstruction of the file.

batch. (1) Pertaining to a program or operation that is performed with little or no interaction between the user and the system. (2) In the ImagePlus system, a stack of documents that are scanned into the system as one group. A batch may consist of a single page, a single document, a multiple page document, or multiple documents of the same type.

before-image. A copy of a block of a record before a modification. (T) Contrast with *after-image*.

binary. (1) Pertaining to a selection, choice, or condition that has two possible different values or states. (I) (A) (2) Pertaining to a system of numbers to the base two; the binary digits are 0 and 1. (A)

binary large object (BLOB). A large stream of binary data whose meaning and interpretation are not pertinent to the discussion at hand, such as the objects stored and cataloged in a library. Synonym for *bit string*.

bind. (1) To relate an identifier to another object in a program; for example to relate an identifier to a value, an address, or another identifier, or to associate formal parameters and actual parameters. (T) (2) The process whereby the output from the DATABASE 2 precompiler is converted to a package.

bit. Either of the digits 0 or 1 when used in the binary numeration system. See also *byte*.

bit map. A coded representation in which each bit, or group of bits, represents or corresponds to an item; for example, a configuration of bits in main storage in which each bit indicates whether a peripheral device or a storage block is available or in which each group of bits corresponds to one pixel of a display image.

bit string. (1) A string consisting solely of bits. (I) (A) (2) The digitized image resulting from a scanned document.

blank. (1) An empty space, one or more characters long. (2) A part of data medium in which no characters are recorded. (A)

BLOB. Binary large object.

block. (1) A string of data elements recorded or transmitted as a unit. The element may be characters, words, or physical records. (T) (2) A collection of contiguous records recorded as a unit. Blocks are separated by interblock gaps and each block may contain one or more records. (A)

BookManager. A family of IBM licensed programs that allow users to create and display softcopy.

break. (1) To interrupt the sending end and take control of a circuit at the receiving end. (2) On an IBM personal computer, to stop operations, by pressing the Control (Ctrl) key and Break key.

browse. (1) To look at information without changing it. (2) To look at records in a file.

buffer. (1) A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another. (A) (2) A portion of storage used to hold input or output data temporarily.

byte. A string that consists of 8 bits, treated as a unit, and representing a character. (T) See also *bit*.

C

cache. (1) A special-purpose storage, smaller and faster than main storage, used to hold a copy of instructions and data obtained from main storage and likely to be needed next by the processor. (T) (2) In a Digital Library system, cache refers to an area of storage that clients use to do the following:

- Store objects retrieved from an object server for users to work with them
- Store objects temporarily before storing them to an object server

caching. Storing instructions and data in a cache.

call. (1) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (I) (A) (2) To transfer control to a procedure, program, routine, or subroutine.

catalog. (1) To create an object from an existing file. After cataloging a file as an object, a user can gain access to it only by using the ImagePlus system. (2) A table relating names to names, objects, or locations of objects.

CD-ROM. Compact disk read-only memory.

character. Any letter, number, ideogram, or other symbol used for the organization, control, or representation of data.

character coded. Pertaining to commands, such as logon or logoff, entered by an end user and sent by a logical unit in character form. The character-coded command must be in the syntax defined in the user′s unformatted system services definition table.

character recognition. The identification of geographic, phonic, or other characters by various means, including magnetic, optical, or mechanical means. (A)

character string. A string consisting solely of characters. (I) (A)

CICS. Customer Information Control System.

CICS/ESA. Customer Information Control System/Enterprise Systems Architecture.

CIF. Common interchange file.

CIU. Common interchange unit.

class. Any category to which things are assigned or defined.

client. (1) A user. (2) A functional unit that receives shared services from a server. (T) See also *client-server*. (3) Clients let users, system administrators, application programmers, and others who have user IDs access and work with the Digital Library system.

client application. An application written with the Digital Library APIs to customize a user interface.

Client Application program. The OS/2-based client application program provided with the Digital Library solution. The Digital Library Client Application program is a complete document management system written with Digital Library APIs. It supports document and folder creation, storage, presentation, processing, and access control. It can be customized with user exits and can be partially invoked with APIs.

client data space. A storage area defined in the volume table of contents of a direct access volume for the exclusive use of a client application.

client-server. The model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

code. (1) One or more computer programs, or part of a computer program. (2) To represent data or a computer program in a symbolic form that can be accepted by a data processor. (I) (A) Synonym for *program*. (3) A set of rules in which code groups are substituted for plain text elements that are primarily words, phrases, or sentences.

code page. (1) An assignment of graphic characters and control function meanings to all code points; for example, assignment of characters and meanings to 256 code points for an 8-bit code, assignment of characters and meanings to 128 code points for a 7-bit code. Each code page is assigned a number or name that defines it and its code-to-character relationships. (2) A particular assignment of hexadecimal identifiers to graphic characters.

collection. (1) A group of objects that typically have similar performance, availability, backup, retention, and class transition characteristics. A collection is used to catalog a large number of objects which, if cataloged separately, could require an extremely large catalog. (2) A group of objects in a management class within a storage group.

column. (1) One of two or more vertical arrangements of lines, positioned side by side on a page or screen. (T) (2) A character position within a print line or on a display. The positions are numbered from 1, by 1, starting at the leftmost character position and extending to the rightmost position.

command. (1) A statement used to request a function of the system. A command consists of the command name abbreviation, which identifies the requested function, and its parameters. (2) A character string from a source external to a system that represents a request for system action.

commit. To cause all changes that have been made to files since the last commitment operation to become permanent, and the records to be unlocked so they are available to other users.

common interchange file (CIF). A file that contains one ImagePlus Interchange Architecture (IPIA) data stream.

common interchange unit (CIU). The independent unit of transfer for a common interchange file (CIF). It is the part of the CIF that identifies the relationship to the receiving database. A CIF may contain multiple CIUs.

compact disk read-only memory (CD-ROM). A high-capacity read-only memory in the form of an optically read compact disk. A CD-ROM is typically made by creating a master and then making duplicates of the master. This is the same as compact disks in the music industry.

compatibility. (1) The ability of two or more systems or components to perform their required functions while sharing the same environment. See also *interoperability*. (2) The capability of a hardware or software component to conform with the interface requirements of a given data processing system without affecting its functions. (T)

compatible. (1) Pertaining to computers on which the same programs can be run without appreciable alteration. (2) Pertaining to the capability of hardware or software to meet the requirements of a specified interface.

compiler. (1) A program that translates a source program into an program that can be run (an object program). (2) A program that translates instructions written in a high-level programming language into machine language.

completion code. A return code indicating that an operation has ended.

component. (1) A hardware or software entity forming part of a system. (2) A piece of logic that controls that operation of a device.

concurrent. Pertaining to processes that take place within a common interval of time during which they may have to alternately share common resources; for example, several programs, when run by multiprogramming in a computer having a single instruction control unit, are concurrent. (T)

condition. (1) One of a set of specified values that a data item can assume. (2) An expression in a program or procedure that can be evaluated as either true or false when the program or procedure is running.

configuration. (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (T) (2) In a Digital Library system, a system configuration specifies all of the components of your Digital Library system, identifies the nodes that you installed the components on, and records the relationships between components. You use the system configuration utility to create a configuration file for your system.

configuration file. A file that specifies the characteristics of a system or subsystem.

For Communications Manager/2, the configuration file contains profiles that you can tailor to control the way your OS/2 node communicates with other nodes.

In a Digital Library system, you use the configuration file for Communications Manager/2 to configure APPN communications.

A configuration file for Communications Manager/2 is different from a system configuration file for a Digital Library system.

configuration server. An OS/2 component of the Digital Library system that acts as an interface between clients and the other components of the Digital Library system. Each configuration server stores profile information about each client that links to it.

connection. (1) In data communication, an association established between functional units for conveying information. (I) (A) (2) In programming, a mechanism that enables interaction among modules, particularly procedure calls to asynchronous procedures. (I)

container. An item that can hold other items. In folder manager, a folder is a container which can contain other folders or documents.

content class. In a Digital Library system, a set of rules that define how the components of an application page—image segment, presentation text segment, and adjunct data—are combined and stored. It is a classification of multiple objects with the same data format. Content class determines the structure of documents and document pages. See also *data format*.

contrast. (1) In a picture rendition, the difference between color or shading of the material of interest and the background on which it is shown. (A) (2) In a monochrome image, the difference between the darkest and the lightest portions.

control. (1) Specification of the time or order in which data is processed. (2) A visual user-interface component that allows a user to interact with data. Controls are usually identified by text; for example, headings, labels in push buttons, field prompts, and titles in windows. (3) Independent means of checking or assuring that work specifications are adhered to.

conversation. A dialog between two transaction programs (for example, across an LU-LU session in SNA LU 6.2 protocol).

convert. To change the representation of data from one form to another, without changing the information it conveys; for example, radix conversion, code conversion, analog to digital conversion, media conversion. (T)

corrective service diskette (CSD). An installation diskette containing corrections or improvements to programs already delivered.

creation date. The date on which some user creates a new data object and adds it to the system.

CSD. Corrective service diskette.

cursor. A visual cue that shows a user where keyboard input will appear on the screen.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

D

DASD. Direct access storage device.

data. (1) A reinterpretable representation of information in a formalized manner suitable for communication, interpretation, or processing. (T)
(2) Any representations, such as characters or analog quantities, to which meaning is or might be assigned. (A)

data field. A component of a record corresponding to an attribute. (T)

data format. The predetermined arrangement of characters, fields, and lines that a Digital Library system uses to store objects. A data format consists of a name and description.

When you add a new data format to a Digital Library system, you must specify the following:

- Path and file name for the software program that handles the data format
- Optional or required parameters to send to the software program

See also *content class*.

data set. The major unit of data storage and retrieval, consisting of a collection of data described by control information to which the system has access.

data space. (1) A storage area defined in the volume table of contents of a direct-access volume for the exclusive use of VSAM to store files, indexes, and catalogs. (2) A storage area defined in the volume table of contents of a direct-access volume for storing data sets, including catalogs.

data stream. (1) A continuous sequence of data elements being transmitted, or intended for transmission, using a defined format. (2) All information (data and control commands) sent over a data link usually in a single read or write operation.

data structure. The syntactic structure of symbolic expressions and their storage allocation characteristics. (T)

data type. (1) In programming languages, a set of values together with a set of permitted operations. (I)
(2) A category that identifies the mathematical qualities and internal representation of data.

database (DB). (1) A collection of data with a given structure for accepting, storing, and providing, on demand, data for multiple users. (T) (2) A collection of interrelated data organized to serve one or more applications. (T)

database management system (DBMS). A computer-based system for defining, creating, manipulating, controlling, managing, and using databases. (T)

database table. In DATABASE 2, a named collection of data consisting of rows and columns.

database utilities. A set of utilities provided with a Digital Library system to perform the following functions:

- Manage workflows
- Release suspended items
- Reorganize the database

DATABASE 2. An IBM relational database management system. DATABASE 2 is used for MVS, DATABASE 2 OS/2 is used for OS/2, and DATABASE 2 6000 is used for AIX.

DB. Database.

DBCS. Double-byte character set.

DBMS. Database management system.

DCA. Document content architecture.

default. (1) A value used by the system when no other value is specified. This value can be set by the system beforehand and can be overridden by a user, or chosen by the system after the user has declined the opportunity to enter or select one. (2) A value provided by or an action performed by a software program for a choice or control when a user presses the Enter key. The program provides default values or actions that users might most likely want in a given situation.

For example, every pop-up window in a Digital Library system includes a default push button, which the program highlights. If users press Enter after completing the information in other parts of the window, the program automatically performs the action represented by the default push button.

default value. Synonym for *default*.

delete. (1) To remove something and close up the space it occupied; for example, to remove the character at the cursor position and shift the characters that are to the right of the cursor one position to the left. (2) A function that enables a user to remove all or part of a previously entered text. (T)

destaging. In a Digital Library system, moving a stored object from an online or higher priority device back to an offline or low-priority device, usually on demand of the system or on request of the user. Contrast with *staging*.

destination. (1) Any point or location, such as a node, station, or a particular terminal, to which information is to be sent. (2) An external logical unit (LU) or application program to which messages or other data is directed.

device. A mechanical, electrical, or electronic contrivance with a specific purpose.

device manager. A collection of routines for complex interfaces that acts as an intermediary between device drivers and computers.

diagnostics. The process of investigating the cause or the nature of a condition or problem in a product or system.

dialog. The interaction between a user and a computer.

dictionary. A database of specifications of data and information processing resources. (A)

Digital Library components. The major components and subcomponents of a Digital Library system are:

- IBM ImagePlus VisualInfo Client for OS/2, Program Number 5621-326
- IBM ImagePlus VisualInfo Client Interfaces for Windows, Program Number 5622-540
- IBM ImagePlus VisualInfo Library Server for OS/2, Program Number 5655-036
- IBM ImagePlus VisualInfo Library Server for AIX, Program Number 5765-258
- IBM ImagePlus VisualInfo Library Server for MVS/ESA, Program Number 5655-071
- IBM ImagePlus VisualInfo Object Server for OS/2, Program Number 5622-213
- IBM ImagePlus VisualInfo Object Server for MVS/ESA, Program Number 5655-072

Digital Library folder. The folder on your OS/2 desktop that contains the Digital Library programs.

Digital Library icon. The icon on your OS/2 desktop that represents the Digital Library folder.

Digital Library system. A Digital Library system can be one of the following:

- A stand-alone system
- A starter system
- A custom system

digitize. (1) In imaging systems, the process of representing images as a matrix of binary code, with 0 representing white space and 1 representing non-white space. (2) In imaging systems, the process of representing images as a bit string where the bit string is a representation of the original document. (3) To express or represent in a digital form data that is not discrete data; for example, to obtain a digital representation of the magnitude of a physical quantity from an analog representation of that magnitude. (I) (A)

direct access. The capability to obtain data from a storage device, or to enter data into a storage device, in a sequence independent from its relative position, by means of addresses indicating the physical position of the data. (T) Contrast with *sequential access*.

direct access storage device (DASD). A device in which access time is effectively independent of the location of the data.

directory. (1) A table of identifiers and references to the corresponding items of data. (I) (A) (2) An index that is used by a control program to locate one or more blocks of data that are stored in separate areas of a data set in direct access storage. (3) A listing of the files stored on a diskette. (4) A container of files and other directories.

disconnect. (1) To disengage apparatus used in a connection and restore it to its ready condition when not in use. (2) To break a connection, physically or electrically.

disk. (1) A round, flat data medium that is rotated in order to read or write data. (T) See *optical disk*. (2) Loosely, a magnetic disk unit.

Disk Operating System (DOS). An IBM licensed program that interacts with the processor in the system unit and the hard disk or diskette drives of the computer to control the flow of data.

display. (1) The visual presentation of data. (I) (A) (2) To present information in a window or on a screen.

DLL. Dynamic link library.

document. (1) A structural unit of text and images that can be stored, retrieved, and exchanged among systems and users as a separate unit. (T) (2) In the ImagePlus system, one or more pages of images, text, or both. (3) In a Digital Library system, a document is an item that can be stored, retrieved, retrieved, and exchanged among systems and users as a separate unit. A single document can contain many different types of base parts, including text, images, and objects such as spreadsheet files. See *object, content class, container, item, BLOB.*

document content architecture (DCA). An architecture that guarantees information integrity for a document being interchanged in an office system network. DCA provides the rule for specifying form and meaning of a document. It defines revisable form text (changeable) and final form text (unchangeable).

DOS. Disk Operating System.

double-byte character set (DBCS). A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Typing, displaying, and printing DBCS characters require special hardware and software support. Contrast with *single-byte character set*.

dynamic. Pertaining to an operation that occurs at the time it is needed rather than at a predetermined or fixed time.

dynamic link library (DLL). A module in a software program that contains a dynamic link routine (DLR) that is linked at load or run time.

E

edit. (1) To enter, modify, or delete data. (2) To modify a numeric field for output by suppressing zeros and inserting commas, periods, currency symbols, the sign status, or other constant information.

element. (1) In a set, an object, entity, or concept having the properties that define a set. (I) (A) (2) In SNA, the particular resource within a subarea that is identified by an element address. (3) A parameter value in a list of parameter values. (4) The smallest unit of data in a table or array.

end user. (1) A person, device, program, or computer system that uses a computer network for the purpose of data processing and information exchange. (T) (2) The ultimate source or destination of information flowing through a system.

enhance. In image processing, to manipulate an original image by algorithms intended to improve its appearance and legibility to make it more easily interpreted by a human. This may include changes to color, gray scale, or bilevel values of portions of the image.

entity. Any concrete or abstract thing of interest, including associations among things; for example, a person, object, or process that is of interest in the context under consideration, and about which data may be stored in a database. (T)

entry. (1) An input operation on a computer. (2) In programming languages, a language construct within a procedure designating the start of the procedure. (I) (3) An element of information in a table, list, queue, or other organized structure of data or control information.

error. A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. (I) (A)

error code. A number returned from a program or called API that classifies a syntax or processing error encountered.

error condition. A state that results from an attempt to initialize instructions in a computer program that are not valid or that operate on data that is not valid.

error log. (1) A data set or file where error information is stored for later access. (2) A form that is used to record error information about a product or system. (3) A record of machine checks, device errors, and volume statistical data.

error message. A short descriptive text indicating that an error has been detected. (A)

ESA. Enterprise Systems Architecture.

event. An occurrence of significance to a task, such as the suspension of an item, or the beginning of the workflow for an item. (A)

exception. In programming languages, an abnormal situation that may arise during run time causing a deviation from the normal run-time sequence. Facilities exist in a programming language to define, raise, recognize, ignore, or handle such a situation.

exit. (1) To initialize an instruction within a portion of a computer program to stop the performance of that portion. Such portions of computer programs include loops, subroutines, modules, and so on. (T) (2) An instruction in a program, routine, or subroutine that causes control to pass to another program, routine, or subroutine. See *user exit*.

expiration date. The date after which an item is deleted or cannot be used.

export. See *import/export*.

extension. Additional equipment on the same line and on the same premises, but at a location other than the main station.

F

facility. (1) An operational capability or the means for providing such a capability. (T) (2) A service provided by an operating system for a particular purpose; for example, the checkpoint and restart facility.

facsimile. Synonym for *fax*.

facsimile (fax) system. A system for transmission of images. The image is scanned at the transmitter, reconstructed at the receiving station, and duplicated on paper.

fax. The process by which a document is scanned, converted into electrical signals, transmitted, and recorded or displayed as a copy of the original.

feature. A part of an IBM product that can be ordered separately by the customer.

FFST. First Failure Support Technology.

field. An identifiable area in a window. Examples of fields are an entry field, into which a user can type or place text, and a field of radio button choices, from which a user can select one choice.

field formatted. Pertaining to a request or response that is encoded into fields, each having a specified format, such as binary codes, bit-significant flags, and symbolic names. Contrast with *character coded*.

file. (1) A collection of information treated as a unit. (2) A named set of records stored or processed as a unit. (T)

fileroom. The location where the Digital Library system stores objects.

filter. A device or program that separates data, signals, or material in accordance with specified criteria. (A) See *scan filter*.

filtering. The process of augmenting the desired density of the image data to reduce other frequencies. Filtering can be used in conjunction with adaptive or fixed thresholding or screening to enhance image quality.

First Failure Support Technology (FFST). An IBM licensed program that improves availability for IBM software applications by providing immediate event notification and first failure data capture for software events, and by automating event tracking and management.

first in-first out (FIFO). A queueing technique in which the next item to be retrieved is the item that has been in the queue for the longest time. (A)

fixed disk. (1) A rigid disk that the user cannot remove from the drive mechanism. (T) (2) A rigid magnetic disk such as the internal disks used in the system units of IBM personal computers and in external hard disk drives.

flag. (1) A variable indicating that a certain condition holds. (T) (2) Any of various types of indicators used for identification.

folder. An item that can contain other folders or documents. In a Digital Library system, a folder can be indexed by key fields. See also container, folder, document.

form. A printed or typed document containing blank spaces for providing requested or required information.

format. (1) A specified arrangement of such things as characters, fields, and lines, usually used for displays, printouts, files, or documents. (2) In programming languages, a language construct that specifies the representation, in character form, of data objects in a file. (I) (3) A specified arrangement of such things as characters, fields, and lines, usually used for displays, printouts, or files.

fully qualified name. A qualified name that is complete; that is, one that includes all names in the hierarchical sequence above the structure member to which the name refers, as well as the name of the member itself.

function. (1) A specific purpose of an entity, or its characteristic action. (A) (2) A subroutine that returns the value of a single variable and that usually has a single exit. (3) A procedure or call included in a program. A function defines and runs a code module.

G

generate. (1) To produce a computer program by selecting subsets from skeletal code under the control of parameters. (A) (2) To produce assembler language statements from the model statements of a macrodefinition when the definition is called by a macroinstruction.

global. (1) Pertaining to all places in a document or file. (2) Pertaining to information available to more than one program or subroutine. Contrast with *local*.

grayed. An indication that a choice in a pull-down menu is unavailable. See *graying*.

graying. An indication that a choice in a pull-down menu is unavailable. For choices that are black when available, graying is accomplished by reducing the contrast between the choice and its background.

group. (1) A set of related records that have the same value for a particular field in all records. (2) A series of records logically joined together. (3) A list of names that are known together by a single name.

group name. A generic name for a collection of input and output devices; for example, disk or tape.

H

handle. A data structure that is a temporary local identifier for an object. Allocating a handle creates it. Binding a handle makes it identify an object at a specific location. (A)

hardcopy. (1) A permanent copy of a display image generated on an output device such as a printer or plotter, and which can be carried away. (T) (2) A printed copy of machine output in a visually readable form; for example, printed reports, listings, documents, and summaries. Contrast with *softcopy*.

header. (1) A block of text printed consistently at the top of one or more pages in a multipage document. A header may contain variable information, such as a page number. (2) A portion of a message that contains control information for the message, such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

Help. (1) A push button that provides information about the item the cursor is on or about the entire window. (2) An action that gives information about the item the cursor is on, an application panel, or the help facility. (3) A menu bar choice that has an associated pull-down menu. Its pull-down menu contains choices that can be requested to invoke help actions.

High Performance File System (HPFS). An OS/2 installable file system that provides fast access to large disk volumes.

highlighting. Emphasizing a display element or segment by modifying its visual attributes. (I) (A)

history. A list of actions performed on a document or within a subsystem.

history log. In a Digital Library system, a file that keeps a record of activities for a workflow.

HPFS. High Performance File System.

I

I/O. Input/output. (A)

IBM. International Business Machines Corporation.

ID. Identifier.

identification. The process that enables recognition of an entity by a system, through personal, equipment, or organizational characteristics or codes. (Data Pro-m)

identifier (ID). (1) One or more characters used to identify or name a data element and possibly to indicate certain properties of that data element. (T) (2) A sequence of bits or characters that identifies a program, device, or system to another program, device, or system.

image. A single page of information in the ImagePlus system; the result of scanning, or digitizing, a single sheet of paper.

image object. (1) An object that contains image data. See *object*. (2) A named bit string. For example, in the context of optical programming support, an image object refers to a scanned paper document.

Image Object Content Architecture (IOCA). A collection of constructs used to interchange and present images.

Image Services. An ImagePlus OS/2 application that provides page capture, document image composition, image page display, and document image transfer to the ImagePlus host system.

import. See *import/export*.

import/export. The sending of information from one system, application, or information resource dictionary (export), and the acceptance of information by another system, application, or information resource dictionary (import). (T)

inactive. (1) Not operational. (2) Pertaining to a node or device not connected or not available for connection to another node or device.

index. (1) A list of the contents of a file or of a document, together with keys or references for locating the contents. (I) (A) (2) In the ImagePlus system, specific information about the document used for folder management and routing information used for workflow management.

index class. A user-defined group of information used to store and retrieve an item or set of items. An index class identifies the type of key fields, automatic processing requirements, and storage requirements for a document or folder.

indexing. (1) The process of assigning a document a unique identifier for eventual storage and retrieval in the system. (2) A process of assigning values to data fields so that documents can be accessed.

indicator. (1) A device that gives a visual or other indication of the existence of a defined state. (T) (2) An internal switch used by a program to remember when a certain event occurs and what to do when the event occurs.

information message. A message that indicates that a condition has occurred that the user can do nothing about. The only actions available to the user are to acknowledge the message or get help. See also *action message*, *warning message*.

initial value. Information in an entry field that is provided by an application when an entry field is first displayed. See also *preselected choice*.

initialization. (1) The operations required for setting a device to a starting state, before the use of a data medium, or before implementation of a process. (T) (2) Preparation of a system, device, or program for operation.

initialize. (1) In programming languages, to give a value to a data object at the beginning of its lifetime. (I) (2) To set counters, switches, addresses, or contents of storage to zero or other starting values at the beginning of, or at prescribed points in, the operation of a computer routine. (A)

inline. In a Digital Library system, an object that is online and in a drive, but has no active mounts. Contrast with *mounted*.

input. (1) Pertaining to a device, process, or channel involved in an input process, or to the associated data or states. (T) (2) Information or data to be processed. (3) One or a sequence of input states. (A) Contrast with *output*.

input data validation. A process used to inspect input data to determine whether it is inaccurate, incomplete, or unreasonable. (T)

input/output (I/O). (1) Pertaining to a device whose parts can perform an input process and an output process at the same time. (I) (2) Pertaining to a functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process. (3) Pertaining to input, output, or both.

insert. (1) To put in or place between. (2) A function or mode that enables the introduction of additional characters within previously entered text. (T)

install. (1) To add a program, program option, or software to a system in such a manner that it is runnable and interacts properly with all affected programs in the system. (2) To connect hardware to a system.

installation. In system development, preparing and placing a functional unit in position for use. (T)

Installation Program. A program that helps the user install or reinstall customized installation files on an ImagePlus computer, as well as the program files, in order to tailor the functions of the computer.

installing. The process of putting a software program on a host system or on a computer.

integrity. The protection of systems, programs, and data from inadvertent or malicious destruction or alteration.

interchange. The capability to import or export an image along with its index from one ImagePlus system to another ImagePlus system using CIF or CIU.

interface. (1) A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes specification of the connection of two devices having different functions. (T) (2) Hardware, software, or both, that links systems, programs, or devices.

internal error. A program error that should be reported to your IBM Support Center for help with problem resolution.

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

interoperability. The ability to interchange data with computers from different vendors without any special conversion or interfacing tools, usually achieved by defining standard formats and protocols.

interpret. To analyze and initialize each statement in a source program before translating and initializing the next statement. (T)

IOCA. Image Object Content Architecture.

IS/2. Image Services.

ISO. International Organization for Standardization.

item. In a Digital Library system, an item is a folder, document, workbasket, or workflow. A Digital Library system lets you specify users′ access to items on an item-by-item basis and index folder and document items. See *folder, document workbasket, and workflow.*

J

job (JOB) statement. A specification that identifies the beginning of a job. It contains such information as the name of the job, account number, and class and priority assigned to the job.

K

key. (1) An identifier within a set of data elements. (T) (2) One or more characters used to identify the record and establish the order of the record within an indexed file.

key fields. The attributes assigned to an index class. Key fields can be used to retrieve a document.

keyword. (1) In programming languages, a lexical unit that, in certain contexts, characterizes some language construct; for example, in some contexts, IF characterizes an if-statement. A keyword normally has the form of an identifier. (I) (2) A name or symbol that identifies a parameter.

L

label. (1) In programming languages, a language construction naming a statement and including an identifier. (I) (2) An identifier within or attached to a set of data elements. (T) (3) A record that identifies a volume on tape, disk, or diskette or that identifies a file on the volume. (4) An identifier of a command generally used for branching.

LAN. Local area network.

LAN-based. Pertaining to software programs that operate in a LAN environment.

language. A set of characters, conventions, and rules that is used for conveying information. (I) (A)

level. (1) The degree of subordination of an item in a hierarchical arrangement. (I) (A) (2) In the ImagePlus system, a programmable convention used to identify transitions between documents as well as transitions between batches of documents.

library. (1) A file or a set of related files; for example, in stock control, a set of inventory control files. (A) (2) A named area on disk that can contain programs and related information (not files). A library consists of different sections, called library members. (3) The set of publications for a product.

library client. The component of a Digital Library system that provides a low-level programming interface for the library system. The library client is a part of the client component of a Digital Library system that resides on a personal computer and runs on OS/2

library object. (1) A named resource stored in a host library system. (2) In the folder manager, a folder, document, or workbasket. It is what the Digital Library system operates on. The three components of a document or folder library object are the index, note, and view windows.

library server. The component of a Digital Library system that contains index information for the objects stored in one or more object servers.

A typical Digital Library system includes one of the following types of library servers:

Library Server for OS/2
: A library server that resides in a personal computer and runs on OS/2

library server for AIX
: A library server that runs in a RISC-based processor on AIX

Library Server for MVS/ESA
: A library server that resides in a System/390 computer and runs on MVS/ESA

Library Server for AIX. An IBM product that is part of the Digital Library solution for image processing and information and library control. The Library Server for AIX is a library server that resides in a RISC-based processor and runs on AIX.

Library Server for MVS/ESA. An IBM product that is part of the Digital Library solution for image processing and information and library control. The Library Server for MVS/ESA is a library server that resides in an IBM System/390 computer and runs on MVS/ESA.

Library Server for OS/2. An IBM product that is part of the Digital Library solution for image processing and information and library control. The Library Server for OS/2 is a library server that resides in a personal computer and runs on OS/2.

licensed program (LP). A separately priced program and its associated materials that bear an IBM copyright and are offered to customers under the terms and conditions of either the Agreement for IBM Licensed Programs (ALP) or the IBM Program License Agreement (PLA).

link. (1) To interconnect items of data or portions of one or more computer programs; for example, linking of object programs by a linkage editor, linking of data items by pointers. (T) (2) Connections between nodes in a Digital Library system. When you create a system configuration for a Digital Library system, you add nodes to the system and then link the nodes together. (3) In SNA, the combination of the link connection and link stations joining network nodes; for example, a System/370 channel and its associated protocols, a serial-by-bit connection under the control of synchronous data link control (SDLC).

list manager server. An OS/2 component in a Digital Library system that controls cache. The list manager server manages electronic queues, or work lists, for the clients that link to it.

The list manager server includes a server process and a dynamic link library (DLL) that contains the application programming interfaces (APIs) for the list manager.

When you create a system configuration for a custom system, you may choose whether to create a node that acts as a list manager server.

The configuration server is a type of list manager server.

load. (1) To feed data into a database. (T) (2) To bring a load module from auxiliary storage into main storage for initialization. (3) To place a diskette into a diskette drive or a magazine into a diskette magazine drive.

load module. A computer program in a form suitable for loading into main storage for initialization.

local. (1) Pertaining to that which is defined and used only in one subdivision of a computer program. (A) (2) Pertaining to a device accessed directly without use of a telecommunications line.
(3) Pertaining to devices that have a direct, physical connection (for example, by cable). Contrast with *global*, *remote*.

local area network (LAN). A computer network located within a limited geographical area. Communication within a LAN is not subject to external regulations. However, communication across the LAN boundary may be subject to some form of regulation.

lock. A serialization mechanism by means of which a resource is restricted for use by the holder of the lock.

log. A record of events that have occurred.

log off. To request that a session be ended.

log on. (1) To initiate a session. (2) In SNA products, to initiate a session between an application program and a logical unit (LU). (3) The procedure for accessing the Digital Library system.

logging. (1) The recording of data about specific events. (2) Recording information into a system log for the purpose of error recovery and tuning.

logical drive. A device that appears to the user as a separate entity but is actually a shared portion of a real device. For example, a hard drive may include several logical drives.

logical unit (LU). In SNA, a port through which an end user accesses the SNA network to communicate with another end user, and through which the end user accesses the functions provided by system services control points (SSCPs).

logical unit (LU) 6.1. A type of logical unit that supports general communication between applications in a host processor.

logical unit (LU) 6.2. A type of logical unit that supports general communication between programs in a distributed processing environment.

logoff. The procedure by which a user ends a terminal session. See *log off*.

logon. The procedure by which a user begins a terminal session. See *log on*.

LU. Logical unit.

M

machine-generated data structure (MGDS). (1) An IBM structured data format protocol for passing character data among the various ImagePlus programs. (2) Data extracted from an image and put into generalized data stream (GDS) format.

management class. Within system-managed storage, a named collection of management attributes describing the retention, backup, and class transition characteristics for a group of objects in a storage hierarchy.

You use management classes to specify when to move objects between the group of storage classes that you assign to the management class.

matching. The technique of comparing the keys of two or more records to select items for a particular stage of processing or to reject unusable records. (T)

member. (1) A partition of a partitioned data set. (2) A data object in a structure, a union, or a library.

memory. All of the addressable storage space in a processing unit and other internal storages that is used to initialize instructions. (T)

menu. (1) A list of choices that can be applied to an object. See *pull-down menu*. (2) A displayed list of items from which a user can make a selection.

menu choice. A choice in a menu that a user can select to apply an action to an object.

message. Information not requested by a user but displayed in response to an unexpected event, or when something undesirable could occur. See also *action message*, *information message*, *warning message*.

message identifier. A 7-character code that identifies a predefined message and is used to retrieve its message description from a message file.

MGDS. Machine-generated data stream.

migrate. To move to a changed operating environment, usually to a new release or version of a system.

migration. (1) The process of moving data and source from one computer system to another computer system without converting the data, such as when moving to a new operating environment.
(2) Installation of a new version or release of a program to replace an earlier version or release.

Minimize. A choice in the title bar pull-down menu that a user selects to remove all windows associated with an application from the screen and replace them with an icon that represents the application.

Mixed Object Document Content Architecture (MO:DCA). An IBM architecture developed to allow the interchange of object data among applications within the interchange environment and among environments.

MO:DCA. Mixed Object Document Content Architecture.

modification. The change or customization of a system, subsystem, or application to work more effectively at a given installation.

module. (1) A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine. (A) (2) A packaged functional hardware unit designed for use with other components. (A)

mount. To place a data medium in a position to operate. (T)

mounted. In a Digital Library system, an object that is online and in a drive, with active mounts. Contrast with *inline*.

Multiple Virtual Storage (MVS). An operating system for large computer systems. The term MVS implies MVS/SP, MVS/ESA, MVS/370, or MVS/XA.

Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA). MVS/ESA is an MVS operating system supporting data spaces and hyperspaces in addition to the 2-gigabyte address spaces of MVS/XA. MVS/ESA is one of the primary operating systems for large computer systems.

MVS. Multiple virtual storage.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture.

N

network. (1) An arrangement of programs and devices connected for sending and receiving information. (2) A configuration of data processing devices and software connected for information interchange.

network identifier (ID). In TCP/IP, that part of the internet address that defines a local area network (LAN). The length of the network ID depends on the type of network class (A, B, or C).

In a Digital Library system, you use the network ID when you create or change a node.

network name. In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), link station, or link.

network table. In a Digital Library system, a text file created by the installation program that records the specific system configuration information for the node you install the Digital Library system on. Each node in the network must have a network table that identifies the node and lists which nodes in the network it can connect to.

The primary purpose of the network table is to map logical server names to the specific communications protocol details required to communicate with the server.

The installation program lets you choose to install one of the following:

- A default network table provided by the Digital Library system for the stand-alone and starter systems
- A customized network table based on the information in the system configuration file that you create using the system configuration utility

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (I) Nodes in an SNA context are processors, controllers, and computers, and are the end point of a link or a junction common to two or more links in a network. Network nodes are interconnected by links. (2) A computer in a network. It is the end point of a link or a junction that is common to two or more links in a network.

In a Digital Library system, a node is either a personal computer with a 386** or higher Intel** processor or a System/390 computer. The node may contain one or more components of a Digital Library system.

node name. The name assigned to a node in a Digital Library system. In a Digital Library system, node names are symbolic names. Symbolic names are names, such as “Node1” or “Node2,” that have some meaning to you.

Each node in the system must have a unique value in the system. In a Digital Library system, this unique value is the combination of the node name and the network identifier (ID).

nonprogrammable workstation (NWS). A workstation that does not have processing capability and that does not allow the user to change its functions.

note. (1) In hardcopy or softcopy documentation, additional information or comments added to a topic or to a line of text in the document. (2) Text annotation to a document or folder. Using notes gives a Digital Library user a way to convey information to other persons who will later use a document or folder.

numeric. Pertaining to data in the form of numerals and some special characters.

O

object. (1) In the Digital Library architecture, objects are data stored in object servers. (2) In Digital Library′s folder manager data model, object specifically refers to a document′s contents or base parts. See *BLOB.* (3) A collection of data referred to by a single name.

object data. A collection of related data elements that have been bundled together. Examples of data elements are graphic characters, image data elements, and drawing orders.

object server. The component of a Digital Library system that contains the objects or information for your end-user applications.

A typical Digital Library system includes either of the following types of object servers:

Object Server for OS/2
    An object server that resides in a personal computer and runs on OS/2
Object Server for MVS/ESA
    An object server that resides in a System/390 computer and runs on MVS/ESA

Object Server for MVS/ESA. An IBM product that is part of the Digital Library solution for image processing and information and library control. The Object Server for MVS/ESA is an object server that resides in an IBM System/390 and runs on MVS/ESA.

Object Server for OS/2. An IBM product that is part of the Digital Library solution for image processing and information and library control. The Object Server for OS/2 is an object server that resides in a personal computer and runs on OS/2.

OCR. Optical character recognition.

offline. (1) Pertaining to the operation of a functional unit that takes place either independently of, or in parallel with, the main operation of a computer. (T) (2) Neither controlled by, nor communicating with, a computer. See also *batch*. (3) In a Digital Library system, an object that is not accessible without manual intervention. Contrast with *online*.

online. (1) Pertaining to the operation of a functional unit when under the direct control of the computer. (T) (2) Pertaining to a user′s ability to interact with a computer. (A) (3) In a Digital Library system, an object that is accessible without manual intervention, but is not in a drive; for example, an object that is in the jukebox but not in the drive. Contrast with *offline*.

operating system (OS). Software that controls the initialization of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. (T)

Operating System/2 (OS/2). Pertaining to the IBM licensed program that can be used as the operating system for personal computers. The OS/2 licensed program can perform multiple tasks at the same time. A Digital Library system requires that you run OS/2 Version 2.1 or higher on its server and computers.

operator. A person who keeps a system running. Contrast with *user*.

optical. The use of light for data transfer involving the use of light-sensitive devices to acquire information for a computer.

optical character. A graphic character printed or hand written according to special rules in order to facilitate an automatic identification by optical means. (T)

optical character recognition (OCR). Character recognition that uses optical means to identify graphic characters. (I) (A)

optical disk. A disk that contains digital data readable by optical techniques. (T)

option. A specification in a statement that may be used to influence the performance of a statement.

order. (1) A specified arrangement used in ordering. (T) (2) To place items in an arrangement in accordance with specified rules. (I) (A)

origin. In a raster image, the picture element (pel) position from which placement and orientation of text, images, and page segments are specified.

OS. Operating system.

OS/2. Operating System/2.

output. (1) Pertaining to a device, process, or channel involved in transferring data outside the system in question or to the associated data or states. (T) (2) Data that has been processed. (3) Data that has been transferred from storage to an output device. Contrast with *input*.

overflow. (1) That portion of the result of an operation that exceeds the capacity of the intended unit of storage. (2) The condition that occurs when the last line specified as the overflow line to be printed on a page has been passed.

overflow volume. A volume that an object server uses for overflow when all other volumes for a storage group are full.

After the object server selects an overflow volume for use, the system changes the status of the volume from **overflow** to **assigned** and associates the volume with that particular storage group.

overlay. A collection of predefined data such as lines, shading, text, boxes, or logos, that can be merged with variable data on a page while printing. See *overlay image*.

overlay image. (1) A pattern such as a report form, grid, or map used as background for a display image. (I) (A) (2) A document image (usually of a standard form) that is used with a coded data overlay.

overload trigger. In a Digital Library system, the maximum number of items allowed in a workbasket before requiring special action. When the overload trigger is exceeded, an overload handler routine is started to control the exception.

override. A parameter or value that replaces a previous parameter or value.

owner. (1) The user or group who creates a profile or is named the owner of a profile. The owner can modify, list, or delete the profile. (2) The user who creates an entity or is named the owner of an entity.

P

page. (1) In the ImagePlus system, the scanned image of one side of a physical piece of paper or similar medium or the information that fills a screen or window. (2) In a Digital Library system, an electronic representation of a piece of paper, stored as image data, coded text data, or a combination of both.

page scan. The electromechanical process of scanning a physical page (paper) to create a bit image of the page.

panel. A particular arrangement of information that is presented in a window or pop-up. If some of the information is not visible, a user can scroll through the information.

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (I) (A) (2) A variable used in conjunction with a command to affect its result.

parameter list. A list of values that provides a means of associating addressability of data defined in a called program with data in the calling program. It contains parameter names and the order in which they are to be associated in the calling and called program.

password. A string of characters known to a computer system and to a user, who must specify the character string to gain access to a system and to the information stored within it.

path. The online storage location of a file. A complete path contains the drive identifier, directory name, and subdirectory name (if any), and the file name and extension. Example:

```
c:\image\myimage\file.ext
```

PC. Personal computer.

performance. One of the two major factors, together with facility, on which the total productivity of a system depends. Performance is largely determined by a combination of throughput, response time, and availability.

permission. (1) Right for a subject to use an access type for a particular object; for example, permission for a process to read a file. (2) The right to use a defined computer resource, such as a library or file.

personal computer (PC). (1) A microcomputer primarily intended for stand-alone use by an individual and possibly connected to mainframes or networks. (T) (2) A desktop, floorstanding, or portable microcomputer that usually consists of a system unit, a display monitor, a keyboard, one or more diskette drives, internal fixed-disk storage, and an optional printer. PCs are designed primarily to give independent computing power to a single user.

Personal System/2 (PS/2). A microcomputer family consisting of desktop, floorstanding, and portable models. Some models are based on the traditional IBM PC architecture and others incorporate Micro Channel architecture, which provides more reliable, efficient data flow, and more direct memory access than the earlier IBM PCs. The PS/2 has built-in functions that once required special adapter boards.

picture. (1) The display image of an area on a document. (2) In a program, a string of characters used in editing to modify the individual characters in a field. There is a one-to-one relationship between the characters in a picture and the characters in its field. (3) A visual representation of graphics or images on a display device.

place holder. In the ImagePlus system, a horizontal bar that identifies the current line on a displayed page. The place holder can be moved up or down on the displayed page.

platform. (1) The operating system environment in which a program runs. (2) In computer terminology, the principles on which an operating system is based.

platter. A medium commonly used with optical storage devices. Platters are generally removable.

pointer. (1) The symbol displayed on the screen that a user moves with a pointing device, such as a mouse. (2) A data element that indicates the location of another data element.

position. (1) Any location in a string that may be occupied by an element and that is identified by a serial number. (T) (2) The location of a character in a series, as in a record, displayed message, or computer printout.

prefix. Characters inserted at the start of a name.

preselected choice. A choice highlighted and selected by the application when a selection field first appears. See also *initial value*.

Presentation Manager. A component of the OS/2 operating system that presents, in windows, a graphics-based interface to applications and files installed and running on the OS/2 program.

print. The action of printing a document or a page at a workstation. The application is initiated using the front-end application or the ImagePlus workstation application.

priority. (1) A rank assigned to a task that determines its precedence in receiving system resources. (2) The relative significance of one job to other jobs in competing for the allocation of resources.

priority value. In the ImagePlus system, a value assigned to an image object each day that it remains in a routing queue. The value is based on the base priority number and the aging priority number.

privilege. A capability that the system administrator for a Digital Library system gives to a user to either access or perform certain tasks on objects stored in the system.

privilege set. A set of privileges for working with system components and functions that you assign to users (user IDs) and user groups.

problem determination. The process of determining the source of a problem; for example, a program component, machine failure, telecommunications facilities, user-installed or contractor-installed programs or equipment, environmental failure such as a power loss, or user error. See also *diagnostics*.

procedure. (1) In a programming language, a block, with or without formal parameters, whose processing is invoked by means of a procedure call. (I) (2) A set of related control statements that cause one or more programs to be performed.

process. (1) A course of events defined by its purpose or by its effect, achieved under given conditions. (2) In data processing, the course of events that occurs during the processing of all or part of a program. (T)

processing. (1) The performance of logical operations and calculations on data, including temporary retention of data in storage while the data is being operated on. (2) The action of performing operations on input data.

processor. In a computer, a functional unit that interprets and initiates instructions. A processor consists of at least an instruction control unit and an arithmetic and logic unit. (T)

profile. (1) Data that describes the significant characteristics of a user, a group of users, or one or more computer resources. (2) In the ImagePlus system, data that provides supervisor and document assignment authorities and defines the type of work to be distributed to the employee from the routing queue.

program. (1) A sequence of instructions suitable for processing by a computer. Processing may include the use of an assembler, a compiler, an interpreter, or a translator to prepare the program for initialization, as well as to run it. (I) (2) In programming languages, a logical assembly of one or more interrelated modules. (I) (3) To design, write, and test computer programs. (I) (A)

programmable workstation. A workstation that allows a user to change its functions.

programmer. A person who designs, writes, and tests computer programs. (A)

programming interface. The techniques identified by a program to allow other programs to request or receive the services of functions provided by that program.

prompt. (1) A visual or audible message sent by a program to request the user′s response. (T) (2) A symbol or action that requests a user entry or selection.

property. A characteristic of an object that can be changed or modified. The properties of an object describe the object. Type style is an example of a property.

protocol. (1) A set of semantic and syntactic rules that determine the behavior of functional units in achieving communication. (I) (2) In SNA, the meanings of and the sequencing rules for requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

PS/2. Personal System/2.

pull-down menu. A list of choices extending from a selected menu-bar choice that gives users access to actions, routings, and settings related to an object.

Q

qualified name. (1) A data name explicitly accompanied by a specification of the class to which it belongs in a specified classification system. (I) (A) (2) A name that has been made unique by the addition of one or more qualifiers.

qualifier. A modifier that makes a name unique. All names in a qualified name are considered qualifiers, except the rightmost, which is called the simple name.

query. (1) A request for data from a database, based on specified conditions; for example, a request for availability of a seat on a flight reservation system. (T) (2) In interactive systems, an operation at a terminal that elicits a response from the system. (3) A request for information from a file based on specific conditions.

queue. (1) A line or list of items waiting to be processed, usually with the rule that items get processed in the order that they have been added to the list; for example, work to be performed or messages to be displayed. (2) To arrange in or form a queue.

R

RACF. Resource access control facility.

random access. An access technique in which logical records are obtained from or placed in a storage device in a nonsequential manner. Contrast with *sequential access*.

read access. The permission to read information.

read-only memory (ROM). (1) A storage device in which data, under normal conditions, can only be read. (T) (2) Memory in which stored data cannot be modified by the user except under special conditions.

record. (1) In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. In some programming languages, records are called structures. (I) (2) A set of one or more related data items grouped for processing.

recover. After a processing failure, to establish a previous or new status from which processing can be resumed. (T)

recovery. (1) The reconstruction of a database; for example, by means of backup files and after-images. (T) See also *after-image*, *backup file*. (2) The resetting of system resources to a point at which computer programs can be restored without error in functional processing. (A)

reference. In programming languages, a language construct designating a declared language object. (I)

refresh. An action that updates the information that a user is currently looking at.

register. A part of internal storage having a specified storage capacity and usually intended for a specific purpose. (T)

release. A distribution of a new product or a new function and fixes for an existing product.

remote. Pertaining to a system, program, or device that is accessed through a telecommunications line. Contrast with *local*.

required field. An entry field or other control in a window in which you must specify a value before you can complete an action.

resolution. A measure of the sharpness of an input or output device capability, as given by some measure relative to the distance between two points or lines that can just be distinguished. A higher resolution makes text and graphics appear clearer.

resource. Any of the data processing system elements needed to perform required operations, including storage, input/output units, one or more processing units, data, files, and programs.

resource access control facility (RACF). A program that provides for access control by identifying and by verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

restore. (1) To return to an original value or image; for example, to return a window to its size and position before a sizing or moving action. (2) To return a backup copy to the active storage location for use. (3) In the ImagePlus system, a process of changing the status of a deleted document to an undeleted document.

retention period. The length of time for which data on a data medium is to be stored. (T)

retrieve. To locate data in storage and move or otherwise prepare it so it can be processed, printed, or displayed. Contrast with *store*.

return. (1) Within a subroutine, to effect a link to the computer program that called the subroutine. (I) (A) (2) In programming languages, a language construct within a procedure designating an end of a processing sequence in the procedure. (I)

return code. (1) A code used to influence how succeeding instructions are run. (A) (2) A value returned to a program to indicate the results of an operation requested by that program.

roll back. To remove changes that were made to database files under commitment control since the last commitment boundary.

rollback. (1) A programmed return to a prior checkpoint. (A) (2) The process of restoring data changed by an application program or user to the state at its last commitment boundary.

ROM. Read-only memory. (A)

route. (1) An established or selected course of action. (2) To send by a selected route or in a specified direction. (3) To prearrange and direct the order and processing of a series of operations.

routine. A program, or part of a program, that may have some general or frequent use. (T)

routing. The process of determining the path to be used for transmission of a message over a network. (T)

row. (1) A horizontal arrangement of characters or other expressions. (A) (2) The horizontal component of a DATABASE 2 table. A row consists of a sequence of values, one for each column of a table.

run. To process a program on a computer; to start.

run time. (1) Any instant at which the initialization of a particular computer program takes place. (T) (2) The amount of time needed for the processing of a particular computer program. (T)

S

save. (1) A function that enables a user to write into a file of a previously entered or modified text. (T) (2) To retain data by copying it from main storage to another storage device.

scale. Making an image, or part of it, smaller or larger. See also *zoom*, *zooming*.

scan. (1) To digitize an image of a document. See also *digitize*, *page scan*. (2) To search records for a specified character string or syntax error.

scan filter. In the ImagePlus system, an optional program that performs some processing on an image after it is scanned such as forms removal, optical character recognition (OCR), and bar code detection. The IPFO program is an example of a scan filter used with a Digital Library system.

search criteria. Key words that are used to retrieve a stored object.

select. The process of explicitly identifying one or more objects to which a subsequent action will apply.

selecting. On a multipoint connection or point-to-point connection, the process of requesting one or more data stations to receive data. (I)

semaphore. An indicator used to control access to a file; for example, in a multiuser application, a flag that prevents simultaneous access to a file.

sequence number. A unique number assigned to an image or document by a scanner to identify or index it in a document image management system.

sequential access. An access method in which records are read from, written to, or removed from a file based on the logical order of the records in the file. Contrast with *direct access*, *random access*.

serial. Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel. (A)

serial number. In a Digital Library system, a unique 8-character identifier assigned to a volume by the manufacturer or when the volume is formatted for use by the operating system. The format is four hexadecimal characters, a colon, and four hexadecimal characters (*hhhh:hhhh*).

server. A functional unit that provides shared services to computers over networks; for example, a file server, a printer server, a mail server. (T) The Digital Library server runs software that supports computers running an image processing application.

session. The period of time during which a user of a terminal can communicate with an interactive system, usually, elapsed time between logon and logoff.

set. A finite or infinite number of objects of any kind, of entities, or of concepts that have a given property or properties in common. (I) (A)

settings. A choice that lets users change the properties of an object.

setup. (1) The preparation of a computing system to perform a job or job step. (2) The configuring, installing, and verification of a program according to a sequence of instructions.

shift. (1) To move data to the right or left. (2) To adjust the contents of a line by moving information to the left or right of its original position.

shutdown. The process of ending operation of a system or a subsystem, following a defined procedure.

significant characters. The characters at the beginning of a file name or field value that must be unique. For example, HPFS supports a non-8.3 file name format, or long file names, but a Digital Library system accepts only the first 11 characters as a file name. Therefore, if you use long file names with HPFS, the first 11 characters should uniquely identify any files that you plan to use with a Digital Library system.

simultaneous. In a process, pertaining to two or more events that occur at the same time, each one handled by a separate functional unit. (T) (A)

single-byte character set (SBCS). A character set in which each character is represented by a 1-byte code. Contrast with *double-byte character set.*

SNA. Systems Network Architecture.

softcopy. (1) A nonpermanent copy of the contents of storage in the form of a display image. (T) (2) One or more files that can be electronically distributed, manipulated, and printed by a user. Contrast with *hardcopy*.

software. All or part of the programs, procedures, rules, and associated documentation of a data processing system. Software is an intellectual creation that is independent of the medium on which it is recorded. (T)

solution. (1) A special program or set of related programs developed to provide a special function or solve a unique problem. (2) The mixture of one component with another to modify the component, create a new component, or modify the system using the component.

source. A system, a program within a system, or a device that makes a request to a target. Contrast with *target*.

source code. Synonym for *source program*.

source module. All or part of a source program sufficiently complete for compilation. (T)

source program. A set of instructions written in a programming language that must be translated to machine language before the program can be run.

space. (1) A site intended for storage of data; for example, a site on a printed page or a location in a storage medium. (A) (2) A basic unit of area, usually the size of a single character. (A) (3) A blank area separating words or lines.

specification. A detailed formulation, in document form, which provides a definitive description of a system for the purpose of developing or validating the system. (T)

SQL. Structured Query Language.

stage. To move a stored object from a relatively slow storage medium to a relatively fast storage medium. Under system-managed storage, the staged object returns to its previous storage location according to the staging time determined by its management class.

staging. In a Digital Library system, moving a stored object from an offline or low-priority device back to an online or higher priority device, usually on demand of the system or on request of the user. Contrast with *destaging*.

staging area. A storage location on the hard drive of the Object Server for OS/2 that temporarily holds copies of objects currently being worked on to reduce access time. The working storage area for the object server.

startup. The initiation of an application.

statement. (1) In programming languages, a language construct that represents a step in a sequence of actions or a set of declarations. (I) (2) In computer programming, a symbol string or other arrangement of symbols. (A) (3) An instruction in a program or procedure.

storage. A functional unit into which data can be placed, in which it can be retained and from which it can be retrieved. (T)

storage class (SC). (1) In system-managed storage, a named list of storage attributes. The list of attributes identifies a storage service level provided for data associated with the storage class. No physical storage is directly implied or associated with a given storage class name. (2) A list of one or more device managers that indicate the type of media associated with a volume. Examples:

- WORM (write once, read many)
- CD-ROM
- Erasable optical

storage class identifier. In system-managed storage, the identifier associated with a storage class. Storage classes are defined during the initial installation of a Digital Library system or during recustomization. This identifier is assigned by the Digital Library system, not by the user.

storage device. (1) A functional unit into which data can be placed, in which it can be retained, and from which it can be retrieved. (I) (A) (2) A facility in which data can be retained. (T)

storage group. (1) In system-managed storage, a named collection of physical devices to be managed as a single object storage area. (2) For OS/2, a set of volumes. For AIX, a set of filesystems. You use a storage group to define a category in which to store groups of objects.

When you define storage groups, you decide the basis for grouping objects together. For example, you might define a storage group that includes a group of similar objects, such as accounting forms, payroll vouchers, or claims forms. You might also define a storage group for objects that users frequently access.

storage group identifier. In system-managed storage, the identifier associated with a storage group. Storage groups are defined during the Digital Library′s initial installation of a Digital Library system, during recustomization, or dynamically by a run-time function. This identifier is assigned by the Digital Library system, not by the user.

storage location. A position in a storage device that is uniquely specified by means of an address. (T)

store. (1) To place data into a storage device. (I) (A) (2) To retain data in a storage device. (I) (A) (3) Deprecated term for storage.

string. A sequence of elements of the same nature, such as characters considered as a whole. (T)

Structured Query Language (SQL). A language that can be used within COBOL, PL/I, APL2, FORTRAN, IBM BASIC, and assembler application programs, or interactively to access DATABASE 2 data and to control access to DATABASE 2 resources.

subsystem. A secondary or subordinate system usually capable of operating independently of, or asynchronously with, a controlling system. (T)

suspend. (1) To set aside or put processing on hold for a document until further information is obtained. (2) In a Digital Library system, to remove an object from its workflow and define the suspension criteria needed to activate it. Activating the object later enables it to continue processing.

suspended item. An item that is set aside to be processed when further information is obtained.

synchronization. The action of forcing certain points in the initialization sequences of two or more asynchronous procedures to coincide in time. (T)

synchronous. (1) Pertaining to two or more processes that depend upon the occurrence of specific events such as common timing signals. (T) (2) Occurring with a regular or predictable time relationship. Contrast with *asynchronous*.

syntax. (1) The relationship among characters or groups of characters, independent of their meanings or the manner of their interpretation and use. (2) The rules for the construction of a statement.

system. (1) In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions. (I) (A) (2) A computer and its associated devices and programs.

system administration program. The program that system administrators use to manage the Digital Library system.

system administrator. (1) The person at a computer installation who controls and manages the use of the computer system. (2) In a Digital Library system, a user with the authority to invoke the system configuration application. The system administrator is also responsible for installation and management of the local area network (LAN).

system configuration. (1) A process that specifies the devices and programs that form a particular data processing system. (2) The manner in which hardware and software of an information-processing system are organized and interconnected. (3) The arrangement and relationship of the components in a system.

system date. (1) The date established for the system when it is started. (2) The date assigned by the system operator during the initial program load procedure.

system ID. The designation or name of a host computer that is specified when a user logs on.

system-managed storage. IBM′s approach to efficient storage management in which the system determines data placement and an automatic data manager handles data backup, movement, space, and security.

system operator. The person who monitors the operation of a computer system and runs the applications and programs on a day-to-day basis.

system programmer. A programmer who plans, generates, maintains, extends, and controls the use of an operating system with the aim of improving overall productivity of an installation.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

The layered structure of SNA allows the ultimate origins and destinations of information, that is, the end users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

T

table. An array of data, each item of which can be unambiguously identified by means of one or more arguments. (I) (A)

table of contents. (1) In hardcopy and softcopy documentation, a list of topics in the order in which they appear in the document. (2) In a Digital Library system, the list of items in a folder or workbasket.

table set. A database configuration that groups application-specific tables together, or groups tables that affect application isolation together.

Tag Image File Format (TIFF). A standard file format that conforms with the TIFF Revision 5.0 specification from Aldus and Microsoft Corporation. A subset of the standard TIFF file formats is supported as a Digital Library content class.

target. Pertaining to a storage device to which information is written.

target system. A system that receives a request from another system to establish communications.

task. In a multiprogramming or multiprocessing environment, one or more sequences of instructions treated by a control program as an element of work to be accomplished by a computer. (I) (A)

text. A graphic representation of information. Text can consist of alphanumeric characters and symbols arranged in paragraphs, tables, columns, and other shapes. Contrast with *graphic*, *image*.

thread. In the OS/2 operating system, the smallest machine instruction to be performed within a process.

threshold. In imaging systems, a level above which all gray-scale image data can be represented as white and below which all gray-scale image data can be represented as black.

TIFF. Tag Image File Format.

time stamp. (1) To apply the current system time. (2) The value on an object that is an indication of the system time at some critical point in the history of the object.

title bar. The area at the top of each window that contains the window title and title bar menu icon. When appropriate, it also contains the minimize, maximize, and restore icons.

token. In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium.

transaction. (1) In a batch or remote batch entry, a job or job step. (2) An exchange between a computer and another device that accomplishes a particular action or result. (3) A specific set of input data that triggers the running of a specific process or job; a message destined for an application program.

transaction code. The short identification name of a transaction.

transmit. To move an entity from one place to another; for example, to dispatch data via a transmission medium or to transfer data from one data station to another via a line. (T)

trigger. (1) To cause immediate initialization of a computer program, often by intervention from the external environment; for example, by means of a manually controlled jump to an entry point. (I) (A) (2) An event or condition that causes a fault to be discovered; for example, abnormal input data or abnormal processing conditions resulting from an overload condition or hardware abnormality.

U

unformatted. (1) Pertaining to that which is not defined, organized. or arranged in a required manner. (2) Pertaining to commands, such as logon or logoff, entered by an end user and sent by a logical unit in character form. Contrast with *field-formatted*.

user. (1) A person who requires the services of a computing system. (2) Any person or any thing that may issue or receive commands and messages to or from the information processing system. (T)

user exit. (1) A point in a program at which a user exit routine may be given control. (2) A programming service provided by an IBM software product that may be requested during the processing of an application program for the service of transferring control back to the application program on the later occurrence of a user-specified event.

user exit routine. A user-written routine that receives control at predefined user exit points. User exit routines can be written in assembler or a high-level language.

user group. A group consisting of one or more individual user IDs defined under a single group name.

user ID. A shortened form of user identification.

user identification (user ID). (1) A string of characters that uniquely identifies a user to a system. (2) The name used to associate the user profile with a user when a user logs on to the system.

user password. A string of characters entered to identify a user to the system.

utility. (1) The capability of a system, program, or device to perform the functions for which it is designed. (2) A computer program in general support of computer processes; for example, a diagnostic program, a trace program, a sort program. (3) A program designed to perform an everyday task such as copying data from one storage device to another.

V

validation. The checking of data for correctness or for compliance with applicable standards, rules, and conventions. (A) See *input data validation.*

value. A quantity assigned to a constant, a variable, parameter, or a symbol.

value set. A set of mutually exclusive, graphical, or textual choices.

variable. (1) In programming languages, a language object that can take different values, one at a time. The values of a variable are usually restricted to a certain data type. (I) (2) A name used to represent data whose value can be changed while the program is running by referring to the name of the variable.

vector. (1) A quantity usually characterized by an ordered set of numbers. (I) (A) (2) In SNA, a data structure containing three fields: a length field that specifies the length of the vector in which it is contained, an identifier or type field, and a value field. The value field can contain subvectors.

verify. To determine whether a transcription of data or other operation has been accomplished accurately. (A)

version. A separate IBM-licensed program, based on an existing IBM-licensed program, that usually has significant new code or new function.

volume. (1) A certain portion of data, together with its data carrier, that can be handled conveniently as a unit. (I) (2) The portion of a single unit of storage accessible to a single read and write mechanism. (3) In a Digital Library system, a representation of an actual physical storage device or unit that the objects in your system are stored in.

Examples:

- A particular floppy diskette
- The D:\ drive on a particular personal computer
- A particular optical disk drive

You can assign volumes to one or more storage groups.

A Digital Library system stores objects in volumes according to your specifications.

volume label. An area on a tape or disk used to identify the tape volume and its owner. On a magnetic tape, this area is the first 80 bytes and contains VOL 1 in the first four positions.

W

warning message. A message that indicates that an undesirable condition could occur, but the user can allow the process to continue. See also *action message*, *information message*.

window. An area of the screen with visible boundaries within which information is displayed. A window can be smaller than or the same size as the screen. Windows can appear to overlap on the screen.

work in process. In the ImagePlus system, an image document or folder that is following a workflow.

work management. A system that lets an enterprise define a work process and environment to automate workflow and control business processes.

workbasket. In a Digital Library system, a collection of documents and folders that are either in process or waiting to be processed. A workbasket definition includes the rules that govern the presentation, status, and security of its contents.

workflow. In a Digital Library system, a sequence of workbaskets that a document or folder travels through while it is being processed.

workstation. One or more programmable or nonprogrammable devices that allow a user to do work. See *nonprogrammable workstation*, *programmable workstation*.

workstation ID. In the ImagePlus system, the identifier of the specific workstation the user wants to use for a session.

WORM. Write-once, read-many.

write access. The permission to write information.

write-once, read-many (WORM). A nonerasable storage medium (for example, an optical disk) that allows part or all of an optical disk to be written on once. The information on the disk can then be read many times.

Z

zoom. To progressively increase or decrease the size of a part of an image on a screen or window.

zooming. The progressive scaling of an entire display image to give the visual impression of movement of all or part of a display group toward or away from an observer. (I) (A)

Index

ITSO Redbook Evaluation

Image and Workflow Library: SmartGuide to
SG24-5159-00

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!)

____

____

____

____

____

Image and Workflow Library: SmartGuide to EDMSuite System Managed Storage

SG24-5159-00

SG24-5159-00
Printed in the U.S.A.