Getting Started with Digital Library V1.0
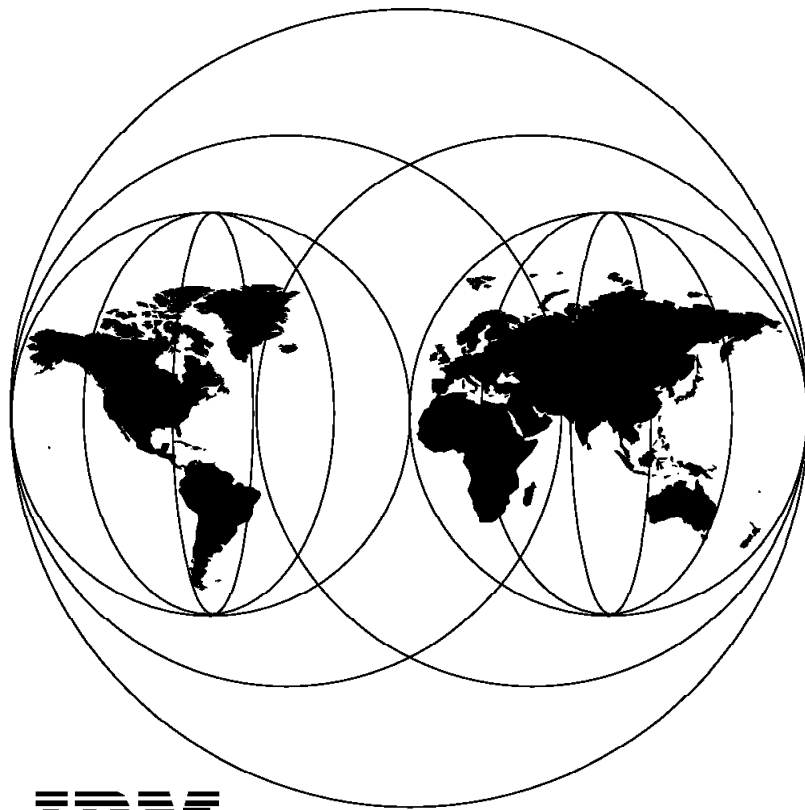
September 1997

IBM

International Technical Support Organization

Getting Started with Digital Library V1.0

September 1997

> **Take Note!**
>
> Before using this information and the product it supports, be sure to read the general information in Appendix C, "Special Notices" on page 89.

First Edition (September 1997)

This edition applies to Version 1.0 of the IBM Digital Library, 84H1437 for use with the operating systems:

- AIX 4.1.5

- OS/2 WARP 3.0

- Windows 95

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

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

Contents

Figures

Tables

Preface

This redbook will help you install and configure IBM Digital Library Version 1.0 focusing on Digital Library Administration and Utilities on AIX, OS/2, and Windows 95. The base installation uses the sample data that comes with the product describing Digital Library solutions with user-modeled data and its Internet connection. However, this redbook will also show you how to customize a solution using a different data model such as the IBM TV commercial videos as the custom-modeled data set.

The Team That Wrote This Redbook

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

Costa Mikhael, IBM Australia.

K.V. Ramakrishnan, IBM India.

Hikaru Tanabe, IBM Japan.

Soon Joo Hyun, ETRI Korea.

We thank the following people for their valuable contributions to this project:

Elsa Barron
International Technical Support Organization, San Jose Center

Mickey Forman
IBM Santa Teresa Laboratory, San Jose

Chris Lee
IBM Santa Teresa Laboratory, San Jose.

Al Madeiros
IBM Santa Teresa Laboratory, San Jose.

Deidra Pacciano
IBM Santa Teresa Laboratory, San Jose.

Tammy Parker
IBM Santa Teresa Laboratory, San Jose.

Philip Sanchez
IBM Santa Teresa Laboratory, San Jose.

Sarah Sherfy
IBM Santa Teresa Laboratory, San Jose.

Pamela Simpson
IBM Santa Teresa Laboratory, San Jose.

Mark Singleton
IBM Santa Teresa Laboratory, San Jose.

Sundar Veliah
IBM Santa Teresa Laboratory, San Jose.

Franklin West
IBM Santa Teresa Laboratory, San Jose.

This project was managed by:

Joe DeCarlo
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 125 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@vnet.ibm.com`

Chapter 1. IBM Digital Library Overview

This chapter presents an overview of the IBM Digital Library (DL), followed by a description of the installation explained in this book. (Digital Library components not used in the installation are also highlighted in this chapter.) We then outline the hardware and software used by this installation. A list of steps is provided which can be used as a guide to installing and configuring all of the Digital Library components. Finally, we briefly explain how to create a custom solution.

1.1 IBM Digital Library

The IBM Digital Library is a software product with many components that can run on multiple platforms. The aim of the product is to provide a system to capture, store, search, distribute, and protect digital information. The product comes with application programming interfaces (APIs) for various components to allow customers to build solutions that meet their specific requirements.

The IBM Digital Library can be configured to run on a single OS/2 machine or scaled up to run on multiple platforms such as AIX, Windows, VM, and OS/2. The type and size of installation will depend on many parameters including the size of library needed, the data access frequency, the type of data and the number of concurrent accesses.

IBM Digital Library Version 1.0 comes with a set of sample data. The installation of an application to be described in this book is called "Sample System." In a real customer environment, once a Digital Library program (such as IBM Digital Library 1.0) is installed, customers need to model their own data and create a client program using the various APIs in order to manipulate the data. The sample system and an example customized system demonstrated in this book show how this can be achieved. The code written for the sample system can be partially reused.

1.2 Sample System

The installation described in this book is one example of a Digital Library installation. Depending on the type and size of solution, your installation will most likely vary in both the number of servers and the number of clients. The installation described here allows you to learn by example. Refer to the Digital Library documentation (on the product CD) for a more comprehensive set of instructions.

The solution described in this document consists of a sample system with the following configuration:

1. Digital Library Administrator (DL Admin) running on OS/2.

2. Library Server with a search manager and the Internet Connection Server for AIX (Cliette) running on an AIX machine.

3. Object server running on a separate AIX machine.

4. Client for Windows running on Windows 95 machine.

Figure 1 on page 2 illustrates the configuration.
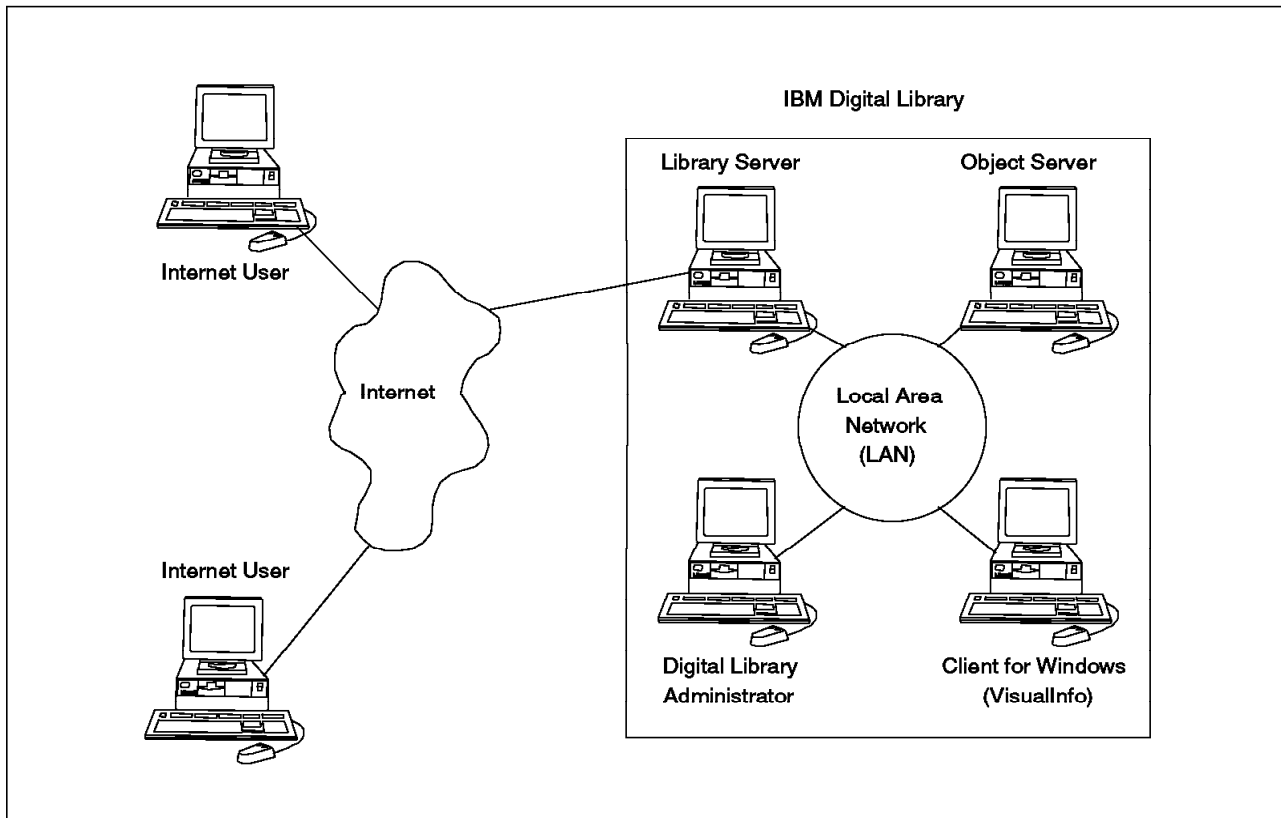
Figure 1. Configuration of the Sample System

The sample system comes with the Digital Library software. The program used to load the data is also provided with the Digital Library software. These components need to be rewritten by the customer when modeling a system other than the sample system.

The installation described in this book does not cover the following Digital Library components:

- System-managed storage (SMS)
- Automated data storage management (ADSM)
- Object Server on platforms other than AIX
- Library Servers on platforms other than AIX
- DL APIs
- List Manager

1.3 Hardware and Software Requirements

This section describes the hardware and software used by this installation. Keep in mind that the requirement is based on installing the sample system.

We use four nodes connected through a token ring network with TCP/IP. We use two AIX machines, a Windows 95 machine and an OS/2 machine. Table 1 gives the detailed hardware and software requirements.

Table 1. Hardware Configuration				
Software Element	Node 1 (DL Admin)	Node 2 (Library Server/Cliette)	Node 3 (Object Server)	Node 4 (Client for Windows)
Machine type	IBM PC 750 166MHZ	RS/6000 36T	RS6000 380	IBM PC 750 166MHZ
Memory	64 MB	128 MB	128 MB	64 MB
Disk Space	2 X 850 MB	2 X 2.0 GB	3 X 4.5 GB	850 MB
Other Peripherals	16 Mbps Token Ring Card	16 Mbps Token Ring Card	16 Mbps Token Ring Card	16 Mbps Token Ring Card

Table 2. Software Configuration				
Software Element	Node 1 (DL Admin)	Node 2 (Library Server/Cliette)	Node 3 (Object Server)	Node 4 (Client for Windows)
Operating System	OS/2 Warp 3.0	AIX 4.1.5	AIX 4.1.5	Windows 95
Prerequisite SW	none	PERL 5.002, DB2/6000 2.1.2, C Set++ 3.1.3, IBM HTTP Server 4.1	DB2/6000 2.1.2, C Set++ 3.1.3	none
IBM Digital Library 1.0	OS/2 DL Admin and Utilities	Library Server, Search Manager, Cliette	Object Server	Client for Windows
DL Patches	None	CSD UR46674, CSD UR47604, Portal Patch (7066)	CSD UR46674	None
Other Patches	none	none	none	none

1.4 Installation Steps

Follow these 22 steps to install the sample system with IBM Digital Library 1.0:

1. Ensure that the proper operating system is installed on each node.

2. Ensure that TCP/IP is properly configured and the four nodes can *ping* each other.

3. Ensure that the language is set to IBM-850 English (United States) on AIX and CODEPAGE=850 in *config.sys* on OS/2.

4. Install Digital Library on OS/2: see 2.1, "Installing DL Admin and Utilities" on page 7.

5. Install requisite software on each AIX node.

   - PERL: see 3.2.1, "Installing PERL and HTTP Software" on page 34.
   - HTTP Server: see 3.2.1, "Installing PERL and HTTP Software" on page 34.

- DB2/6000: see Appendix B, "Installation of DB2/6000" on page 87.
- C Set++: see A.5, "Installation of Any Software" on page 84.

6. Install Library Server, Search Manager, Cliette on AIX (Library Server node): see Chapter 3, "Library Server and Cliette" on page 29.

7. Install Object Server on AIX (Object Server node): see 4.1, "Install Object Server for AIX" on page 41.

8. Install Digital Library Patches for both Object Server and Library Server: see A.6, "How to Install Fixes" on page 85.

9. Install Client for Windows: see 5.2, "Digital Library Client for Windows Installation" on page 52.

10. Configure Digital Library nodes and generate a configuration file on OS/2: see 2.2, "Digital Library System Configuration" on page 12.

11. Configure Library Server on OS/2: see 2.4, "Library Server Configuration from OS/2" on page 18.

12. Configure and run Object Server: see 4.2, "Configure and Run Object Server for AIX" on page 43.

13. Configure and run Library Server: see 3.1.2, "Generating the Library Server Setup File" on page 31.

14. Configure and run Client for Windows: see 5.2, "Digital Library Client for Windows Installation" on page 52.

15. Configure Search Manager on OS/2: see 2.5, "Search Manager Configuration from OS/2" on page 21.

16. Configure and run Search Manager on AIX: see 3.2.2, "Installing and Configuring Search Manager" on page 34.

17. Load sample data (Sample1) on AIX: see 3.2.4, "Loading Sample Data" on page 38.

18. Perform Search Manager indexing for Sample1 on OS/2: see 2.6, "Loading Sample Data" on page 26.

19. Load Sample2 on AIX: see 3.2.4, "Loading Sample Data" on page 38.

20. Perform Search Manager indexing for Sample2 on OS/2: see 2.6, "Loading Sample Data" on page 26.

21. Start Cache Manager and Connection Manager on AIX: see 3.2.5, "Viewing Sample Data" on page 38.

22. Test using a Web browser: see 2.7, "Testing Sample System" on page 26.

1.5 Creating a Custom Solution

The last chapter of this book describes the steps that must be followed for creating a Digital Library solution with custom-modeled data other than the sample data. We define the new data model and describe the changes to be made to the program of the sample system in order to store and search the user-modeled data.

The main purpose of this solution is to give you a greater understanding of the type of changes required when designing your own Digital Library solution. Since each customized solution will most likely imply a new data model, and possibly new requirements for accessing the data, it is recommended that the Digital

Library Application Programming Guide be followed when implementing your own complete solution.

The chapter describes a Digital Library solution with user-modeled data and its Internet connection design guide. For demonstration purpose, the new solution uses IBM Digital Library 1.0 with IBM TV commercial videos as the custom-modeled data set.

Chapter 2. Digital Library Administration and Utilities on OS/2

This chapter describes the steps required to install the Digital Library Administration (DL Admin) and Utilities on OS/2. These components will allow us to define the Digital Library configuration.

Facilities for defining the Indexes and Classes used by Search Manager and the Library Server on AIX are also described in this chapter. Finally, in this chapter, the Sample System is loaded and tested using a Web browser.

Ensure that the OS/2 machine meets the Digital Library hardware and software requirements.

You also need to ensure that TCP/IP is running correctly and that the hostnames of the various Digital Library nodes are defined either using Domain Name Service (DNS) or in a local file. Digital Library uses hostnames and not TCP/IP addresses to communicate in the OS/2 environment. The hostnames can be entered via the **tcpcfg** OS/2 command. Use **ping <hostname>** to find out if a connection can be established. In our installation, the following hosts are defined locally:

```
libsrvr: 9.112.33.167
objsrvr: 9.112.35.28
dlclient: 9.112.32.167
dladmin: 9.112.35.29 (local OS/2 machine)
```

2.1 Installing DL Admin and Utilities

Here are the steps required for installing the DL Admin and Utilities on OS/2:

1. Insert the IBM Digital Library 1.0 CDROM into the OS/2 CD drive.

2. Go to the OS/2 System and select the CD drive from the drives application. In our case, this is **E.**

3. Select the **DLOS2** directory on the CD drive and run the install.exe program.

4. The first screen displayed should be as shown in Figure 2 on page 8
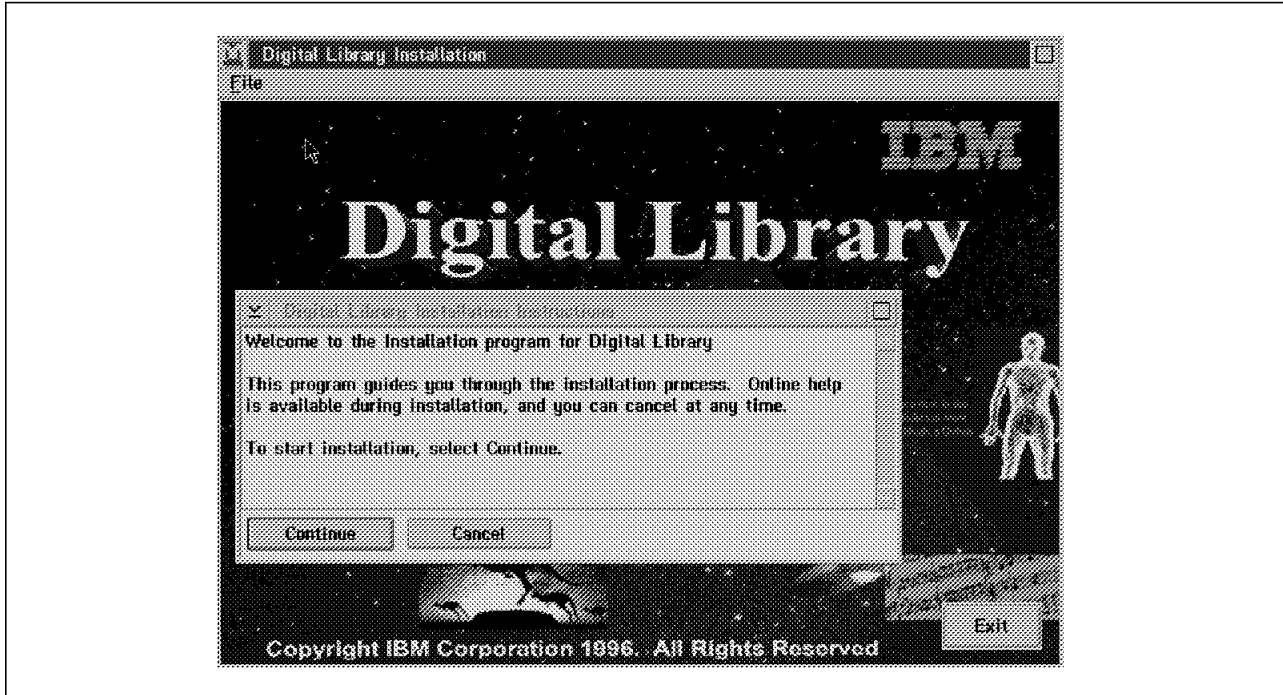
Figure 2. Digital Library Welcome Screen

5. Click **Continue** following the welcome message to start your installation.

6. This opens the Installation and Maintenance Window as shown in Figure 3.
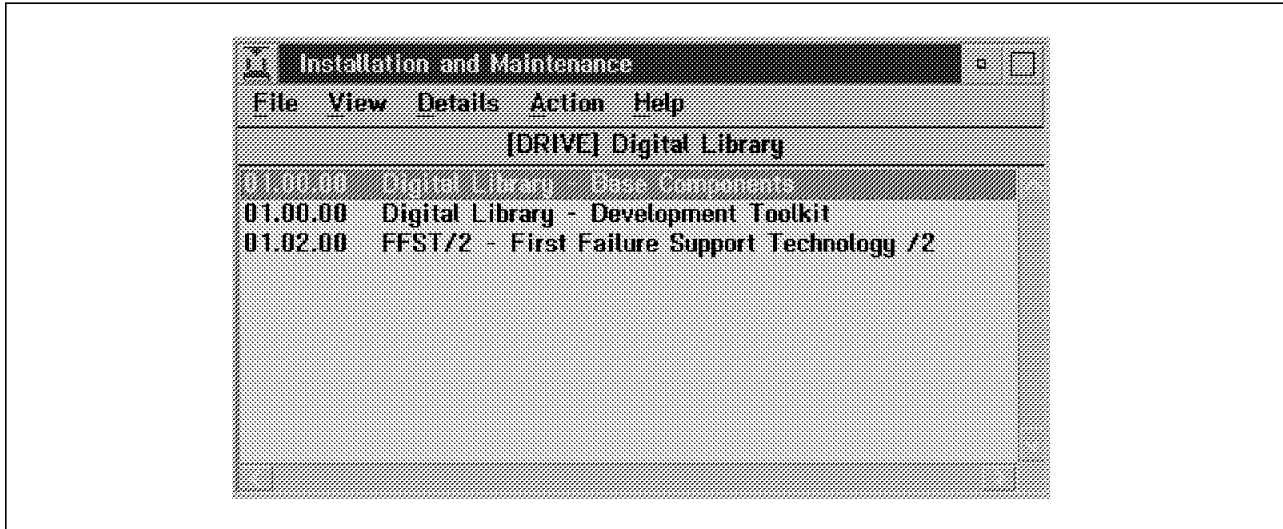
Figure 3. Digital Library Base Component

For our installation, we install only the first option, Base Component. You select this option by double-clicking on the item in the list;

```
01.00.00  Digital Library - Base Component
```

7. Next, you get a dialog indicating the product number and an option to update *CONFIG.SYS*. Make sure that this option is selected and click **OK**.

8. The next window should be as shown in Figure 4 on page 9.
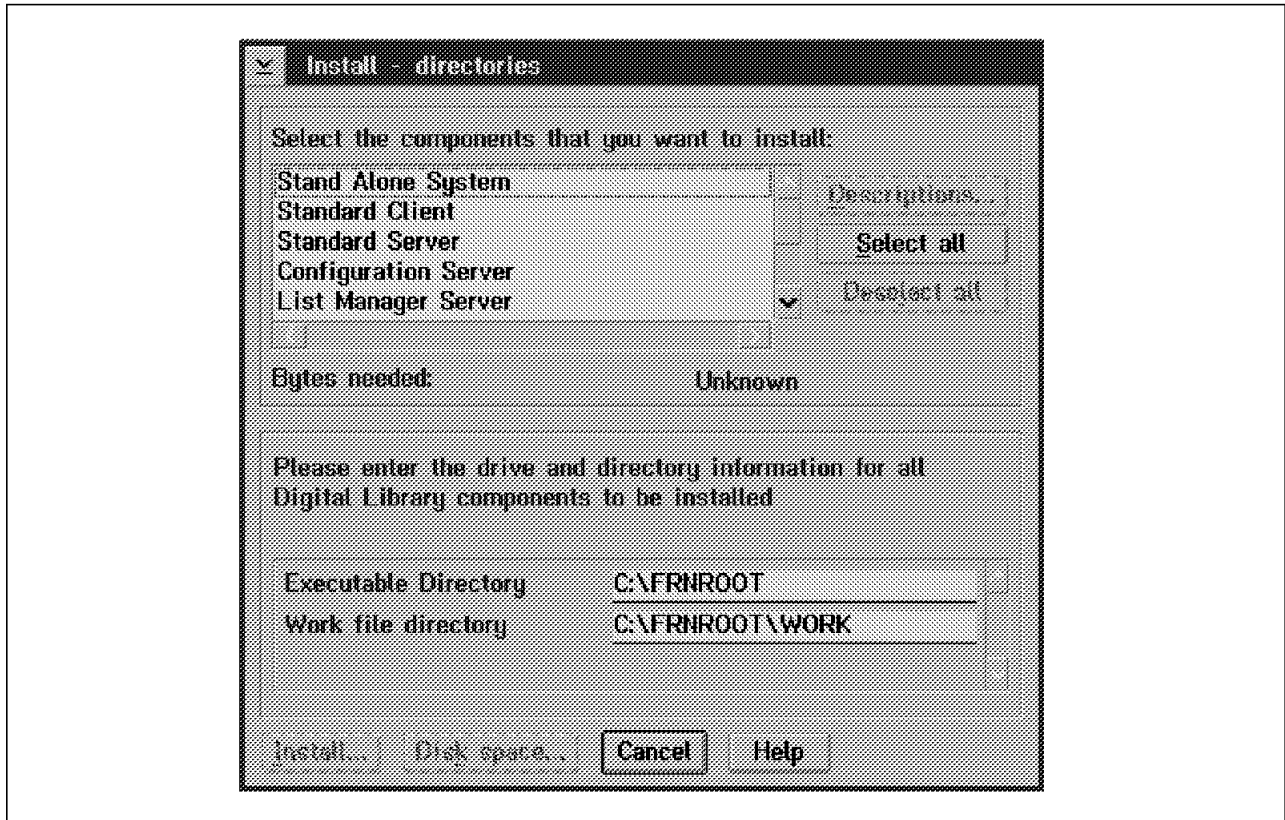
Figure 4. Digital Library Components

For our purpose, we need to install only the DL Admin utilities; therefore, we select the following:

```
Configuration Server
System Configuration Utility
Library Client/Folder Manager/Sys Admin
```

Before selecting **Install...** you may click the **Descriptions...** button to find out the definition of each of the selected items. Also, before selecting **Install...** ensure that enough disk space is available on the selected installation directory.

9. Select **Install...** and wait while watching the installation status window. After about 2 minutes you should see the window shown in Figure 5 on page 10.
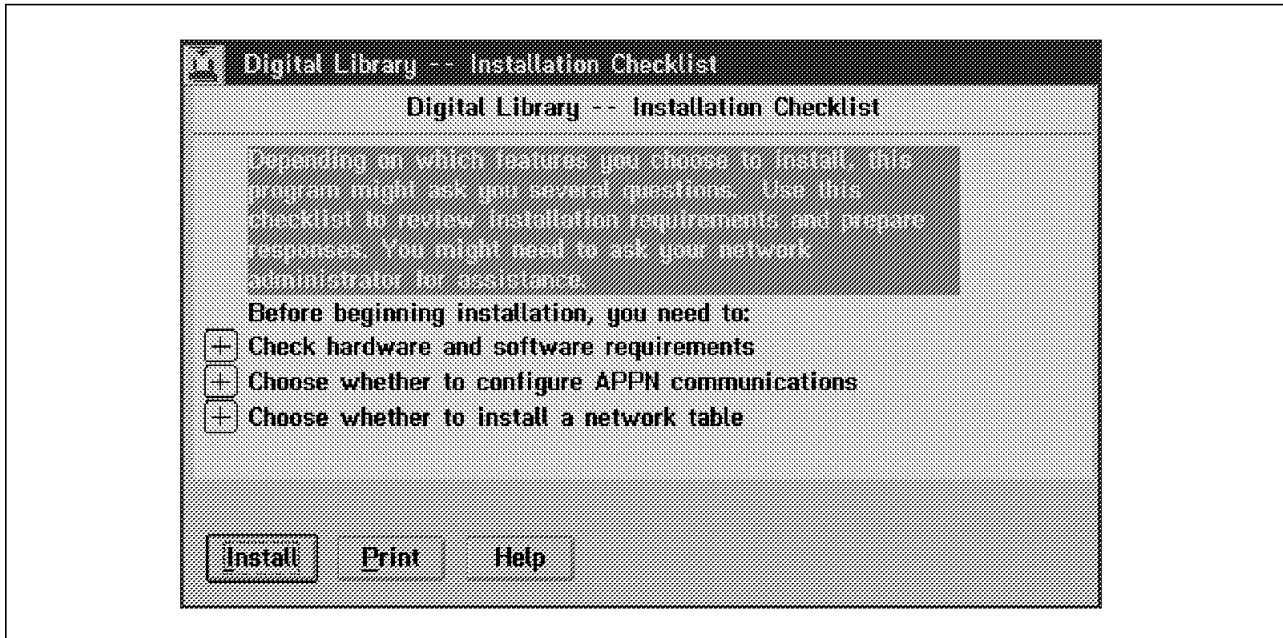
Figure 5. Digital Library Installation Checklist

Click the **Install** button. There is nothing to change here. You should see the window as shown in Figure 6.
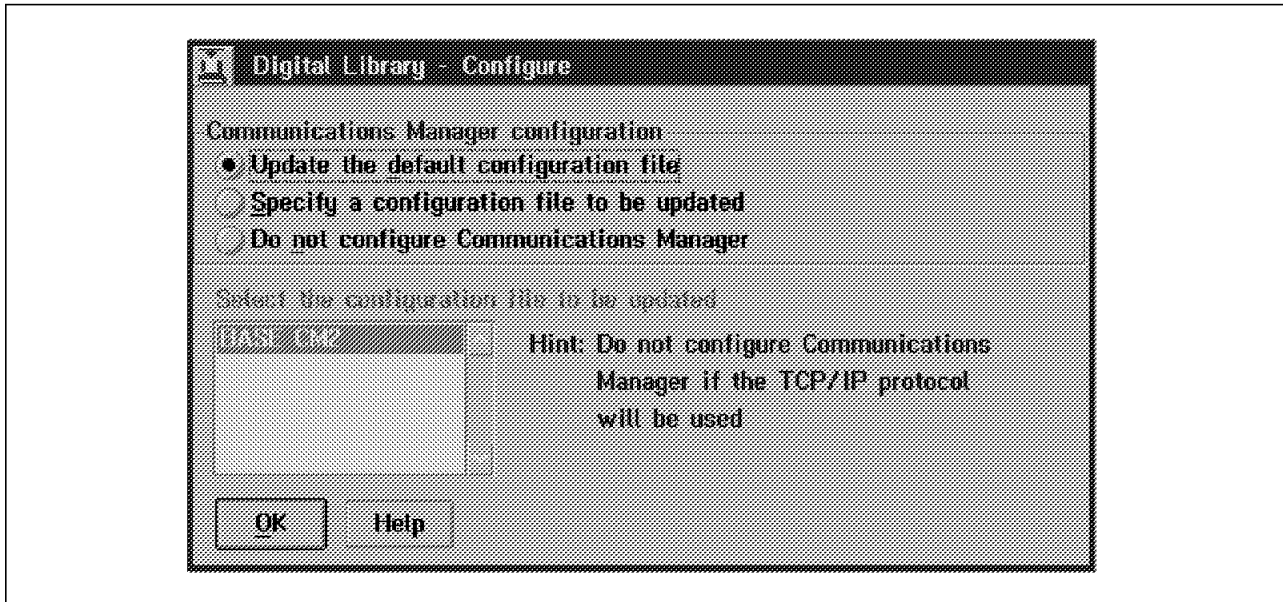
Figure 6. Digital Library Communication Manager

We are using TCP/IP, so the third item,

Do not configure Communications manager

should be selected. At this point, a window appears as displayed in Figure 7 on page 11.
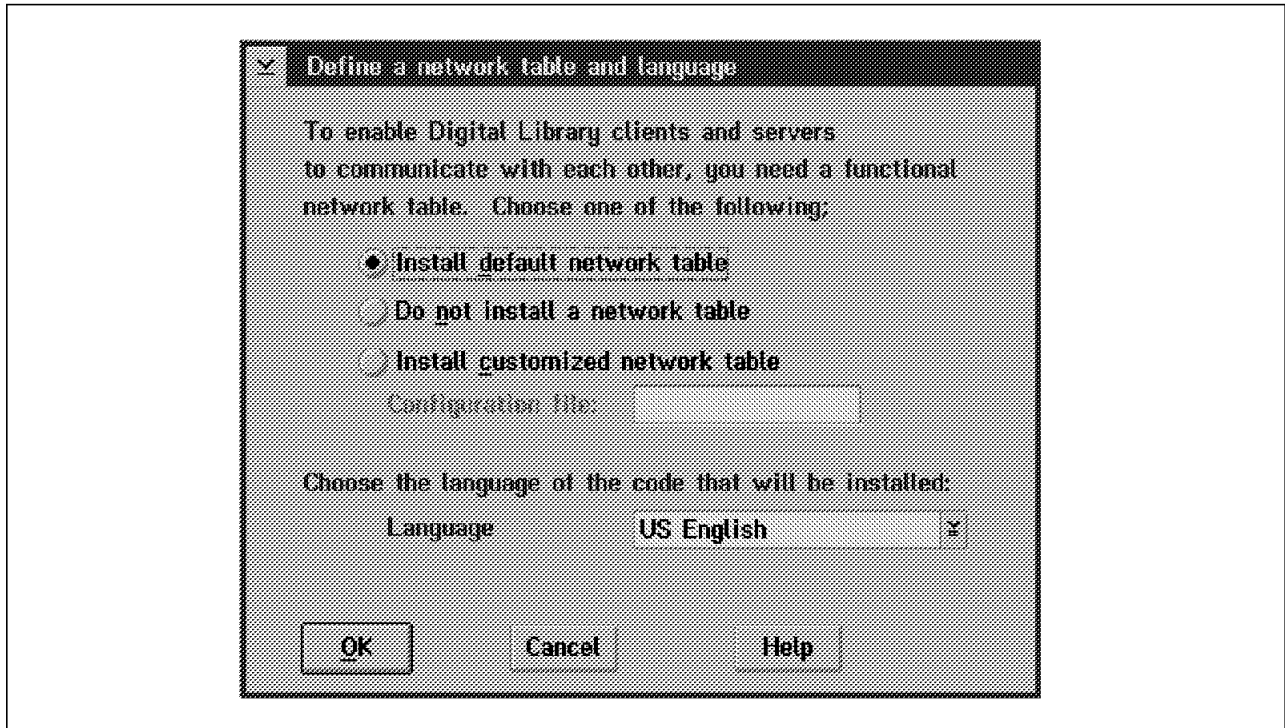
Figure 7. Digital Library Network Table

> For our purpose, leave everything as is, which means **Install default network table** and the Language is set to **US English**.

10. In the end, you should get the window shown in Figure 8. At this point, just click **OK**. A new Digital Library folder should be created.
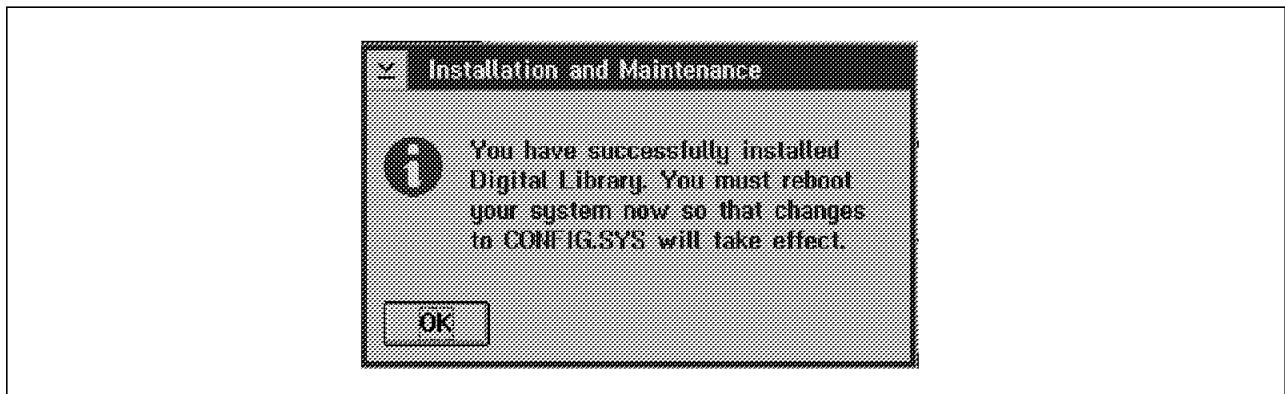
Figure 8. Final Installation Window - Changing Config.Sys

11. Installation is complete. You now need to close the installation window. It does not close automatically on completion.

12. As the last step, you need to set codepage to 850 in *config.sys*. Edit *config.sys*, enter the following line, and then reboot the machine:

```
CODEPAGE=850
```

2.2 Digital Library System Configuration

In this section, we start the Digital Library configuration. Before proceeding, ensure that the TCP/IP hosts are set up correctly. Use ping to ensure that the IP link is up between the OS/2 and each of the hosts.

Following the installation of DL Admin on OS/2, the first step to start the configuration is to create the system configuration file. This file is used as input when generating a network table on each machine. Follow the steps given below to create the system configuration file:

1. Double-click on the **Digital Library** icon. You should now see the folder as shown in Figure 9.

Figure 9. Digital Library Folder

2. Double-click on **Digital Library Utilities** icon. The **Digital Library Utilities Icon View** should open as shown in Figure 10.
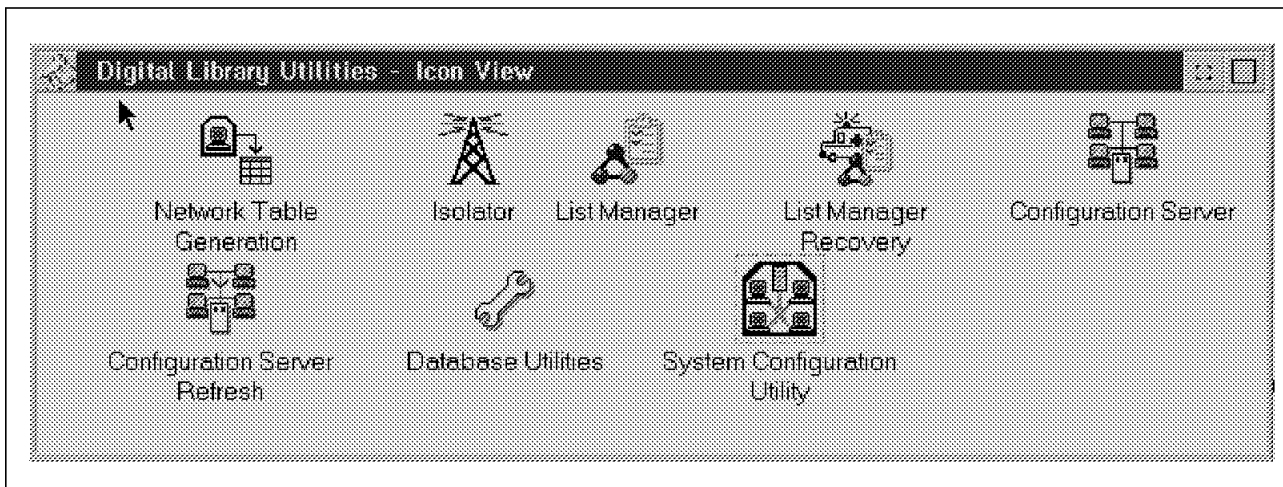
Figure 10. Digital Library Utilities

3. Double-click on **System Configuration Utility** icon. The following folder should appear, as shown in Figure 11 on page 13
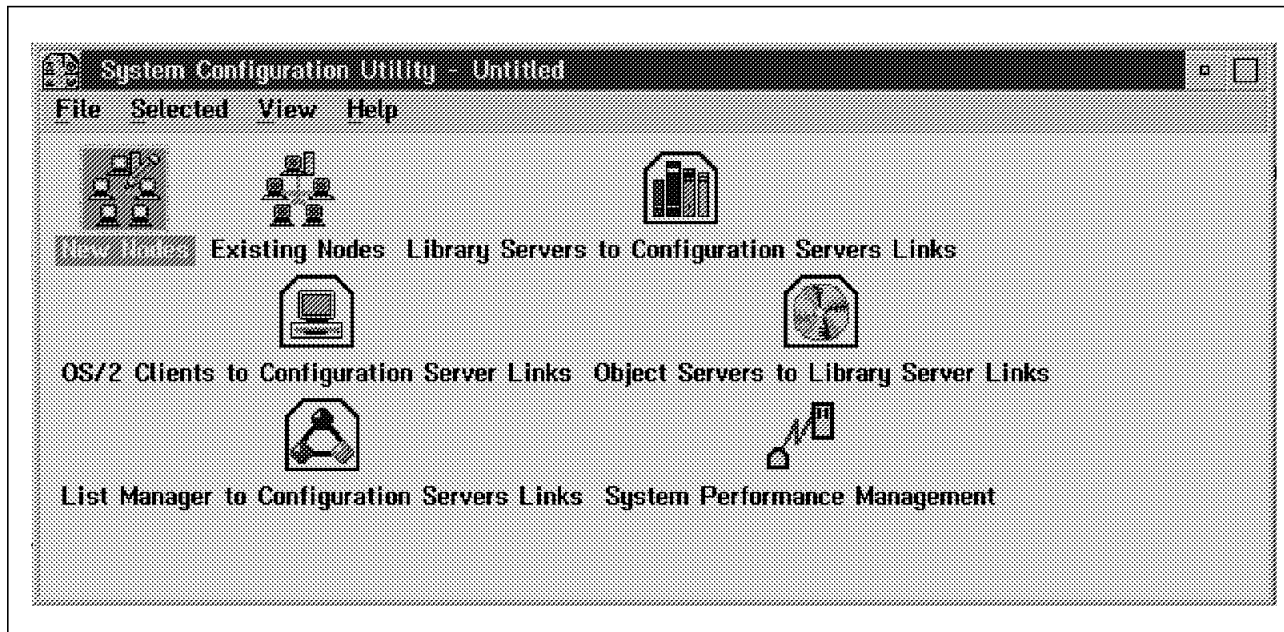
Figure 11. Digital Library Configuration Utilities

4. Double-click on **New Nodes** icon. The following screen should appear, as shown in Figure 12 on page 14.
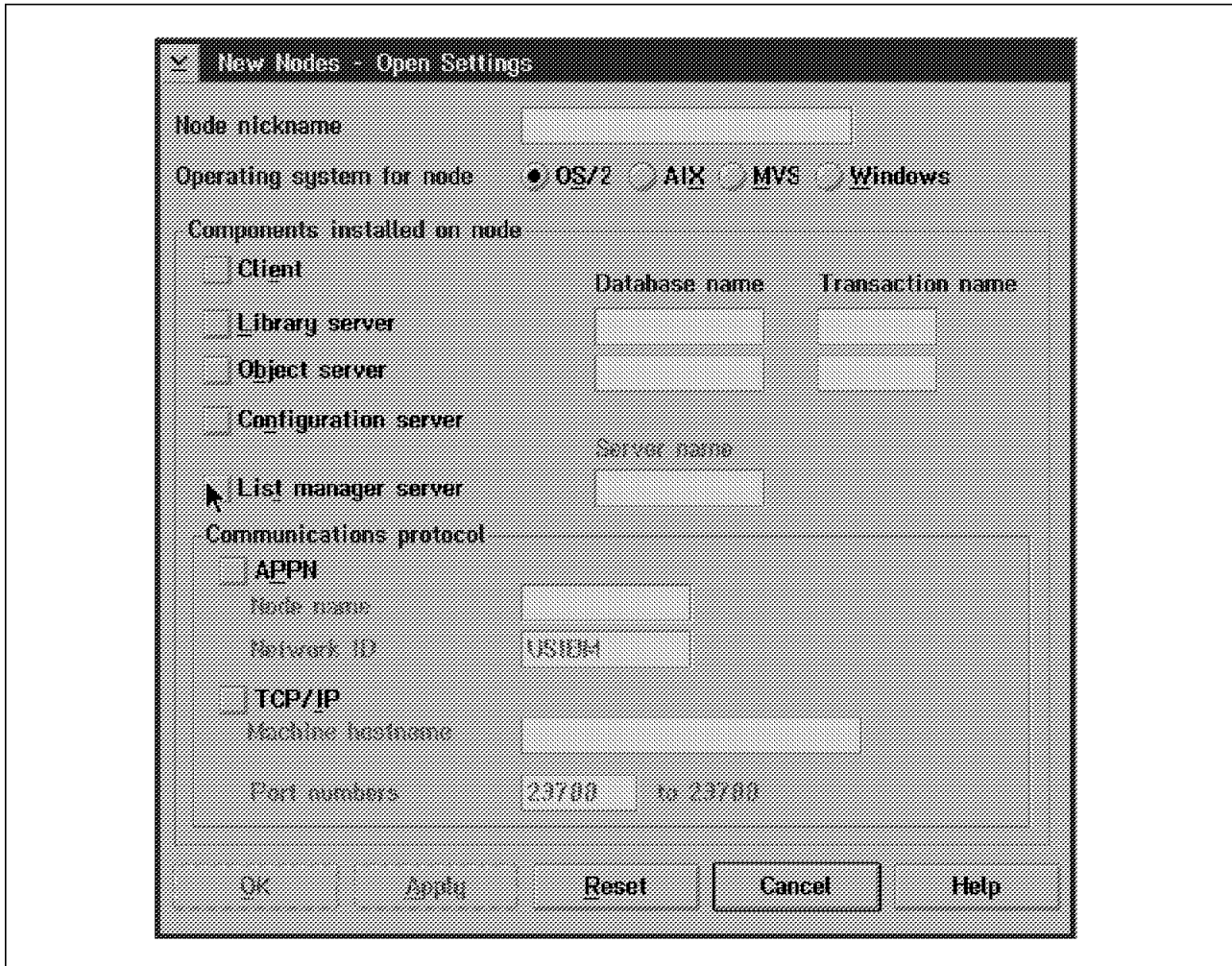
Figure 12. Defining a New Node

5. You need to define a new node for each node in your Digital Library system. In our case we need to define four nodes: The parameters for defining these nodes are outlined in Table 3 Click on the **Apply** button after defining each node.

Table 3 (Page 1 of 2). Defining new Digital Library nodes				
Item	**Node 1**	**Node 2**	**Node 3**	**Node 4**
Node nickname	dladmin	dlclient	objsrvr	libsrvr
Operating system for node	OS/2	Windows	AIX	AIX
Component installed on node	Configuration server/Client	Client	Object server	Library server
Database name	(blank)	(blank)	OBJSRVRX	LIBSRVRX
Transaction name	(blank)	(blank)	(blank)	(blank)
Communication protocol	TCP/IP	TCP/IP	TCP/IP	TCP/IP

Table 3 (Page 2 of 2). Defining new Digital Library nodes				
Item	**Node 1**	**Node 2**	**Node 3**	**Node 4**
Machine hostname	dladmin	(blank)	objsrvr	libsrvr
Port numbers	23788	(blank)	23790	23800

When defining the last node, click **OK** instead of **Apply**. If you hit **Apply** you get another blank window; just quit that window by closing it or by hitting **Cancel**. You should now be back in the **System Configuration Utility** Window. Also, note that the nickname need not be the same as the host name. In our installation we kept them the same.

6. In this step, you need to set up the Library Servers to Configuration Servers links. This is done by double-clicking on the **Library Servers to Configuration Servers links** icon. You should see a window as shown in Figure 13.
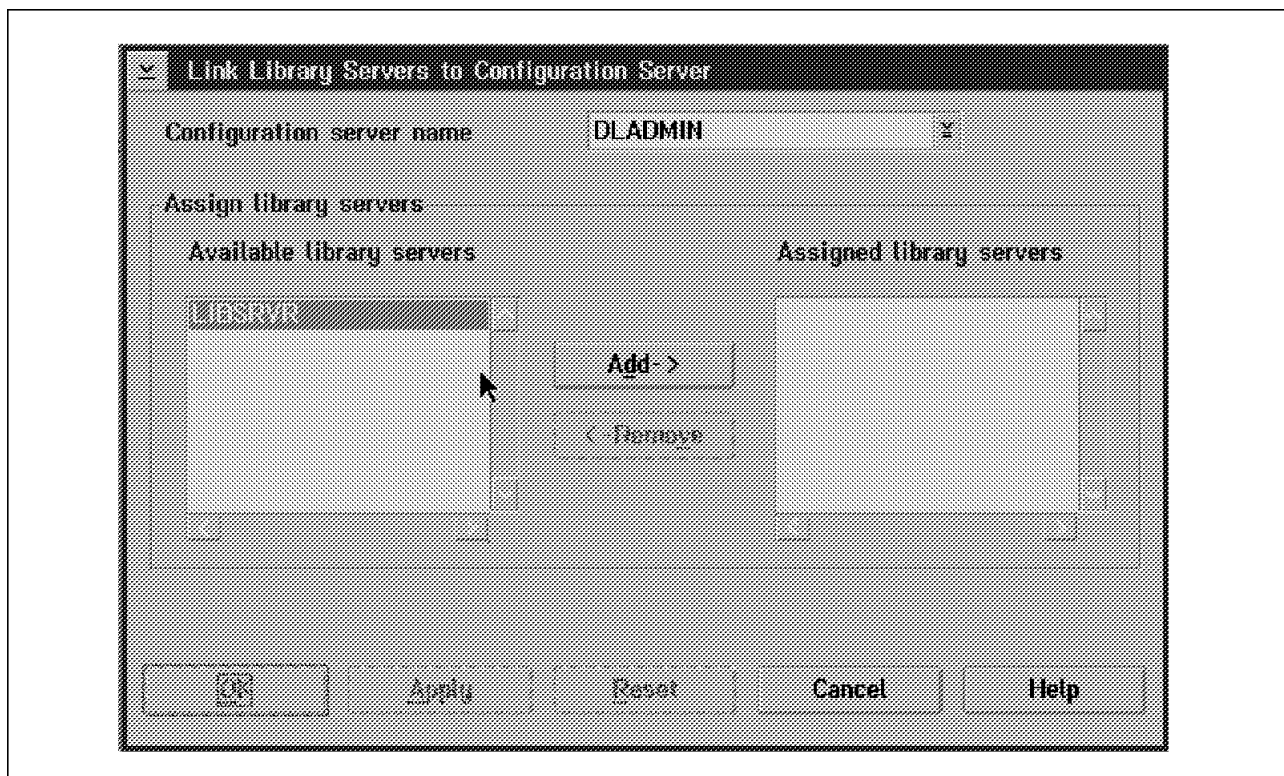
Figure 13. Library Server Configuration

Click **Add->** and LIBSRVR moves to the Assigned library servers list. Hit the **OK** button to complete this operation.

7. The next step is to set up the OS/2 Clients to Configuration Server Links. This is done by double-clicking on the **OS/2 Clients to Configuration Server Links** icon. Select **DLADMIN** from the Available Clients and then hit **Move->** to move it to the Assigned Clients. Then click **OK**

8. In this step you need to set up the Object Servers to Library Servers links. This is done by double-clicking on the **Object Server to Library Server links** icon. You should see a window as shown in Figure 14 on page 16.
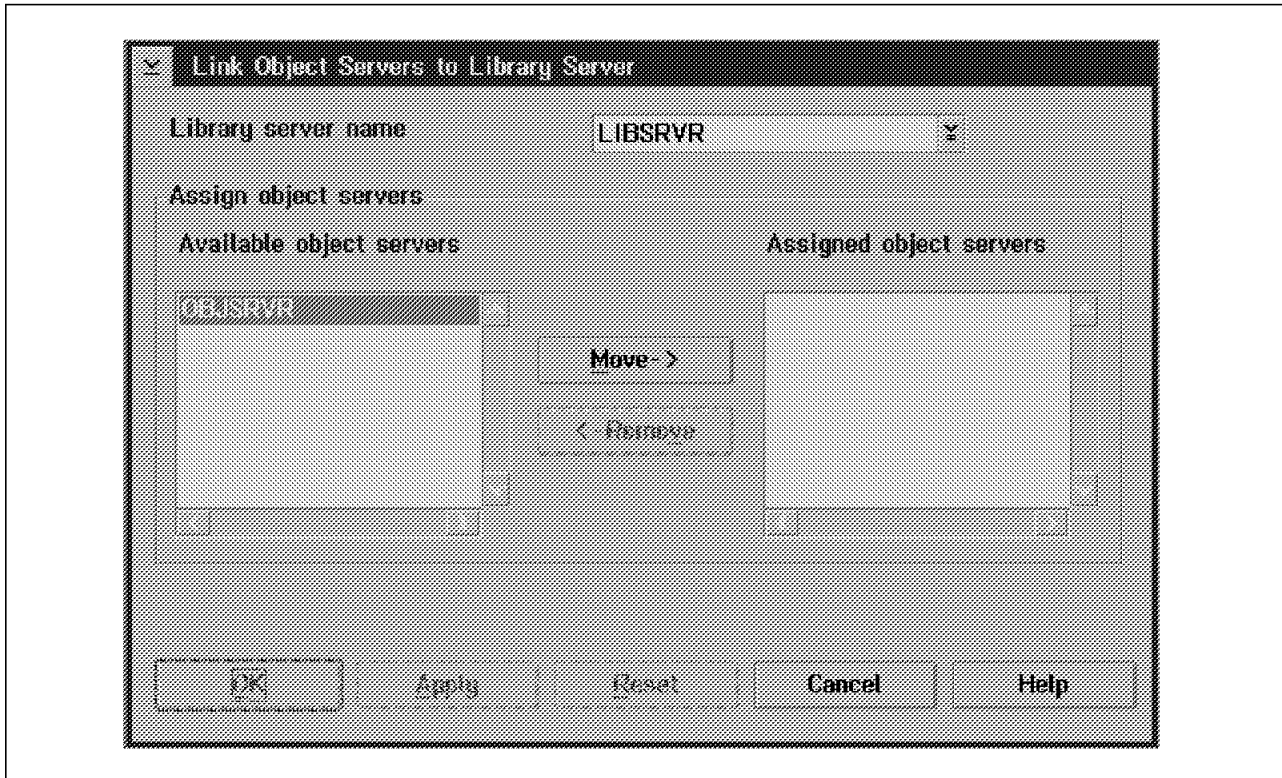
Figure 14. Object Server Configuration

> Click the **Move->** button and you should see OBJSRVR move to the
> Assigned object servers list. Hit the **OK** button to complete this operation. In
> this example, List Manager to Configuration Servers Links and System
> Performance Management were not touched.

9. In the **System Configuration Utility** window, from the File menu select **Verify**.
   You should now get a message saying the verification was successful. If the
   verification was not successful, then read the message in the log file
   \frnroot\frnoutnc.log. This file contains information on the cause of the error.
   If an error occurs, you may need to repeat one or more of the above steps
   until verification is successful.

10. Once verification is successful, then save the configuration file by selecting
    the **Save** option from the File menu. Give this file any name, but remember
    the name; you need it to generate the network table in the next step. We
    named it *dl.cfg*

2.3 Generating Network Table

Follow these steps to generate the network table file for OS/2 DL Admin.

1. In the **Digital Library Utilities** window, double-click on **Network Table
   Generation**.

2. Enter the System Configuration Utility file-name generated in the previous
   steps. In our example it is *\frnroot\dl.cfg*. You are then prompted to enter
   the node nickname for OS/2; in our case, this was *dladmin*. You should now
   get a message in an **OS/2** window indicating that the network table has been
   successfully created.

Note: The network table generation utility creates the network table file, *FRNOLINT.TBL*, and the Digital Library shutdown command file, *FRNOSTOP.CMD*. Both are stored in the *\FRNROOT* directory. Here is what our network table file looks like:

```
/* Network Table Generation Utility                              */
/* DLADMIN/.                                                     */
/* Mon Apr 14 13:49:22 1997                                      */

SERVER: OBJSRVRX REMOTE SOCKETS
            HOSTNAME      = objsrvr
            PORT          = 23802
        SESSION_TIMEOUT = 0
            SERVER_TYPE   = FRNOSAIX

SERVER: OBJSRVRX.REXE REMOTE SOCKETS
            HOSTNAME      = objsrvr
            PORT          = 23804
        SESSION_TIMEOUT = 5000
            SERVER_TYPE   = FRNOX

SERVER: OBJSRVRXADMIN REMOTE SOCKETS
            HOSTNAME      = objsrvr
            PORT          = 23803
        SESSION_TIMEOUT = 0
            SERVER_TYPE   = FRNOSADM

SERVER: LIBSRVRX REMOTE SOCKETS
            HOSTNAME      = libsrvr
            PORT          = 23790
        SESSION_TIMEOUT = 0
            SERVER_TYPE   = FRNLSAIX

SERVER: LIBSRVRX.REXE REMOTE SOCKETS
            HOSTNAME      = libsrvr
            PORT          = 23791
        SESSION_TIMEOUT = 5000
            SERVER_TYPE   = FRNLX

SERVER: FRNOCSVR LOCAL LAN_WIDE SOCKETS
            PORT          = 23788
            PATH          = \FRNOLMLM.EXE
            START         = WINDOW
            ICON          = \FRNOLMLM.ICO
            PASSNAME
            SERVER_TYPE   = FRNCS

SERVER: FRNCACHE LOCAL LOCAL_ONLY
            PATH          = \FRNOLMLM.EXE
            START         = WINDOW
            ICON          = \FRNOLMLM.ICO
            PASSNAME
            SERVER_TYPE   = FRNCACHE

ALIAS: FRNONSVR FRNOCSVR

SERVER: FRNODAOS LOCAL LAN_WIDE SOCKETS
            PORT          = 23788
```

The next step is generating network table files for other nodes. Each node needs to generate its own network table. The two AIXs (for **libsrvr** and **objsrvr**) need the configuration file generated on the OS/2 (*\frnroot\dl.cfg*) as input to their network table generation utility. Use file transfer protocol (FTP) to transfer this file across to the other machines. Ensure that binary mode is used during the FTP transfer (refer to 3.1.3, "Generating a Network Table File" on page 31 for information on generating the Network Table for AIX).

2.4 Library Server Configuration from OS/2

These steps follow the successful installation of the Object Server and Library Server. Ensure that both servers are running before starting the following steps:

1. In the Digital Library folder, double-click on **System Administration**. You will need to log on to the Library Server. The window shown in Figure 15 is displayed.
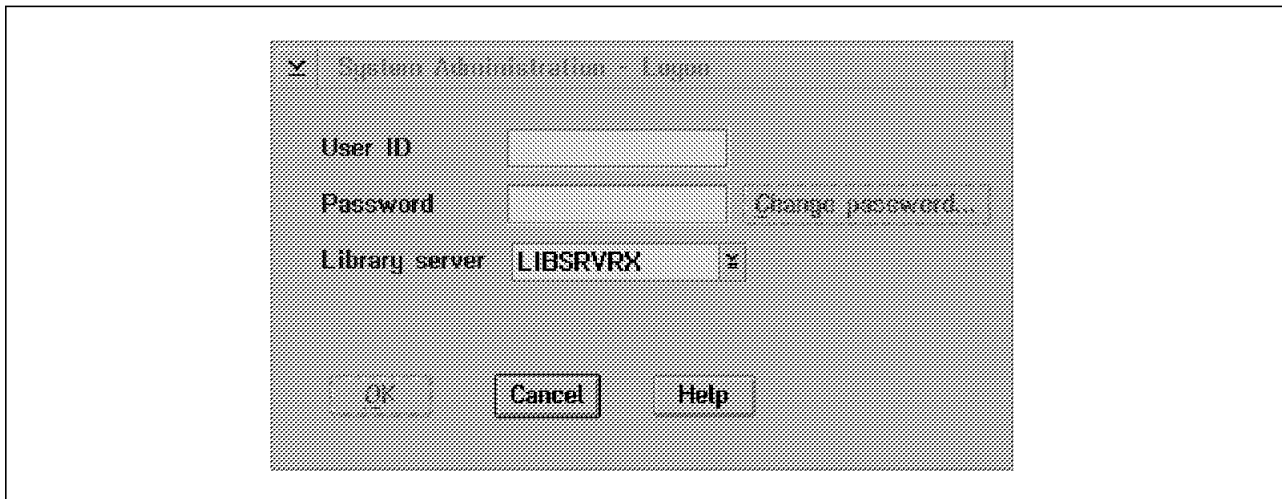
Figure 15. Digital Library Installation Logon

The User ID is *FRNADMIN* and the password is *PASSWORD*. Leave the Library Server as LIBSRVRX. Successful logon should result in the window shown in Figure 16.

Figure 16. Digital Library Installation

Click on **OK**.

2. The System Administration folder now opens. Double-click on **Library Server Configuration**. This opens a new folder, as shown in Figure 17 on page 19.
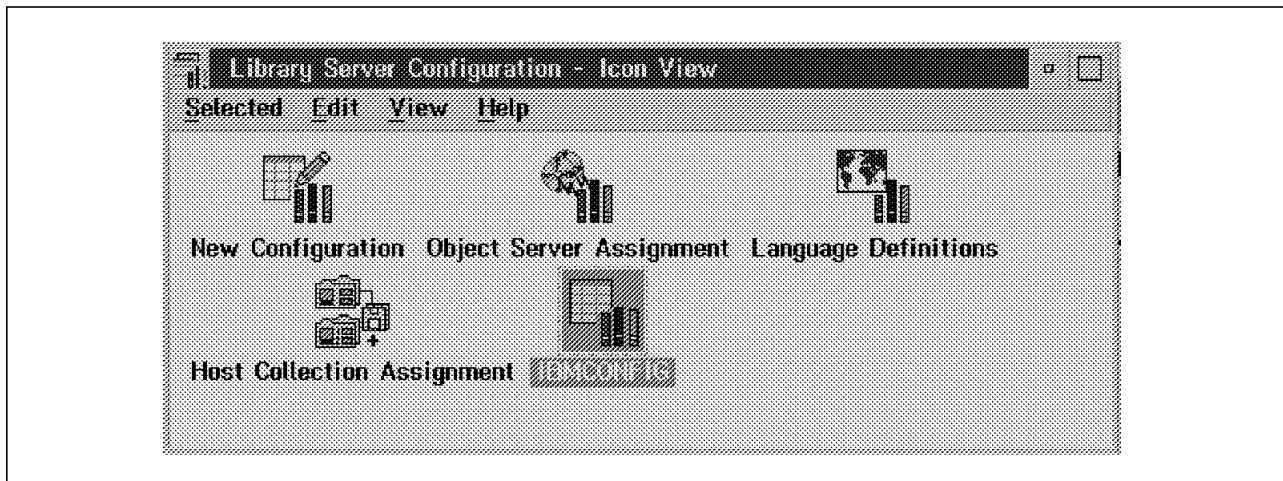
Figure 17. Library Server Configuration

3. In this step, you need to copy the **IBMCONFIG** icon to NEWCONFIG. With the mouse click the right button on the **IBMCONFIG** icon and then when the pop-up menu comes up, use the left button to select **Edit** and then **Copy**. You should now select a new config name; we call it **NEWCONFIG**. A copy of the **IBMCONFIG** icon called NEWCONFIG, now appears.

4. Double-click on **NEWCONFIG**. A window that allows you to specify the setting will open. The window is as shown in Figure 18 on page 20.
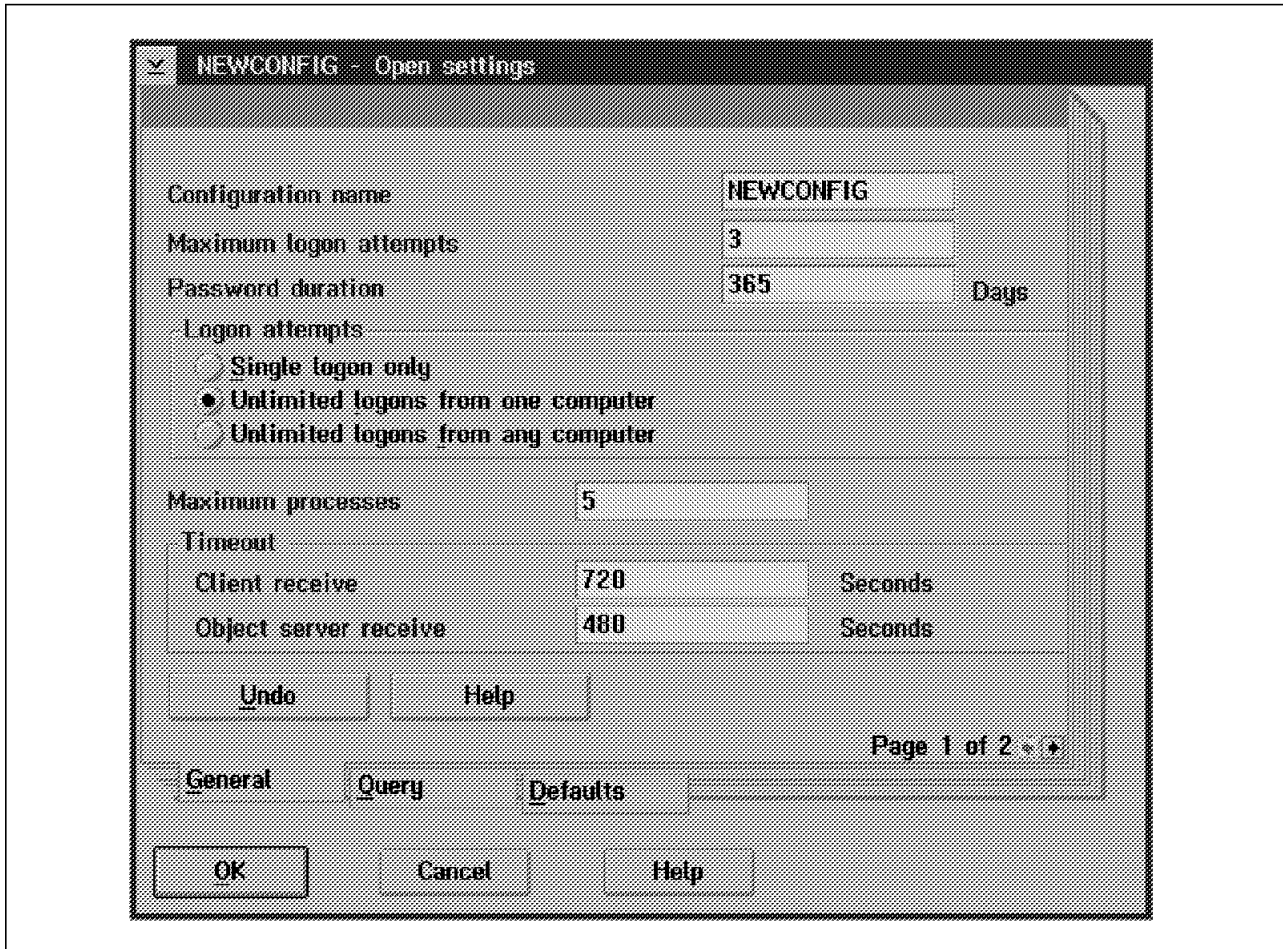
Figure 18. Library Server Configuration—NEW CONFIG

The parameters in the **NEWCONFIG** window should be set as follows:

- Configuration name: **NEWCONFIG** (set as default)

- Maximum logon attempts: **3** (set as default)

- Password duration **365 days** (set as default)

- Logon attempts: change to **Unlimited logons from any computer** (not a default)

- Maximum processes: change to **10**

- Timeout

  – Client receive **720 seconds** (set as default)

  – Object server receive **480 seconds** (set as default)

Click on **OK**.

5. Close both the library server configuration window and the system administration window. Double-click on **Digital Library Services Shutdown** to shut down the Digital Library. Also issue an **frnstop** command to stop both the object server on objsrvr and the library server on libsrvr (AIX).

2.5 Search Manager Configuration from OS/2

Follow the steps in 3.2.2, "Installing and Configuring Search Manager" on page 34 to configure SM6000 on AIX and then come back here. These are the steps to configure the search manager:

1. Configure the SM6000 client. From an OS/2 window, change directory to *c:\frnroot*. Type ehscxmnt at the command line and press ENTER. (If your Library Server is on an OS/2 machine, replace the **ehscxdsp** command with **ehscxmnt** throughout the directions in this step.) An explanation of the capabilities and syntax appears with the **ehscxmnt** command. Refer to this whenever you need to know the capabilities and syntax of **ehscxmnt**.

2. Enter the following command:

   ehscxmnt /u:c /s:DLSAMPLE /n:DLSAMPLE /m:nnn.nnn.nnn.nnn /c:tcpip /p:7502,

   where nnn.nnn.nnn.nnn represents the IP address of the AIX machine that the SM6000 server runs on. In this installation, this is libsrvr and the command is:

   ehscxmnt /u:c /s:DLSAMPLE /n:DLSAMPLE /m:9.112.33.167 /c:tcpip /p:7502.

   Successful execution will be indicated by the message, "Created Search Service Entry: DLSAMPLE."

   Note: The value assigned to the /p: parameter must match the -sp port parameter used in the *smconfig* script in 3.2.2, "Installing and Configuring Search Manager" on page 34. It must also match the port assigned in the ID_APPID= parameter in the *frncliet.ini* file on the AIX machine that the SM6000 server runs on the Library Server. Each of these values should be 7502.

3. Start the Library Server on libsrvr. Refer to 3.1.5, "Starting the Library Server" on page 32 for starting the Library Server.

4. Start SM6000 on the Library Server (AIX). Refer to 3.2.3, "Starting and Stopping Search Manager" on page 37 for starting SM6000.

5. From the same OS/2 window, run the frnodefs program. Then, respond to the queries with the following values:

   - Library Server Name: LIBSRVRX (This is the library server database name.)

   - User ID: FRNADMIN The user ID that SM6000 server uses to log on to the Digital Library is the same user ID you entered into the *excdlpwu* utility on the AIX machine that runs SM6000 server.

   - Password: PASSWORD.

   Successful execution will be indicated by a series of messages that show index parameters being created and that ends with a message:

   Search index class created successfully

   If you get a failure, then check that SM6000 and the Library Server are running properly.

6. Creating Indexes:

   a. Ensure that SM6000 and the Library Server are still running. Do the following: Double-click on the **System Administration** icon. Log on to Digital Library with user ID=FRNADMIN and Password=PASSWORD.

b. Double-click on **Text Search Administration**. Then double-click on **Index Creation**. In the **Create New Document Index window**, enter the following data:

- Search Service Name: DLSAMPLE
- Title: DLSEARCH
- Description: (blank)
- Type: Linguistic
- Name: DLSEARCH
- Location for Index files /usr/lpp/SM6000/index/dlsearch
- Index work file /usr/lpp/SM6000/work/dlsearch
- Do not make any changes in the Document Library fields.

Click on **OK** button. Successful completion will return a message, "Index Created Successfully" (refer to Figure 19).
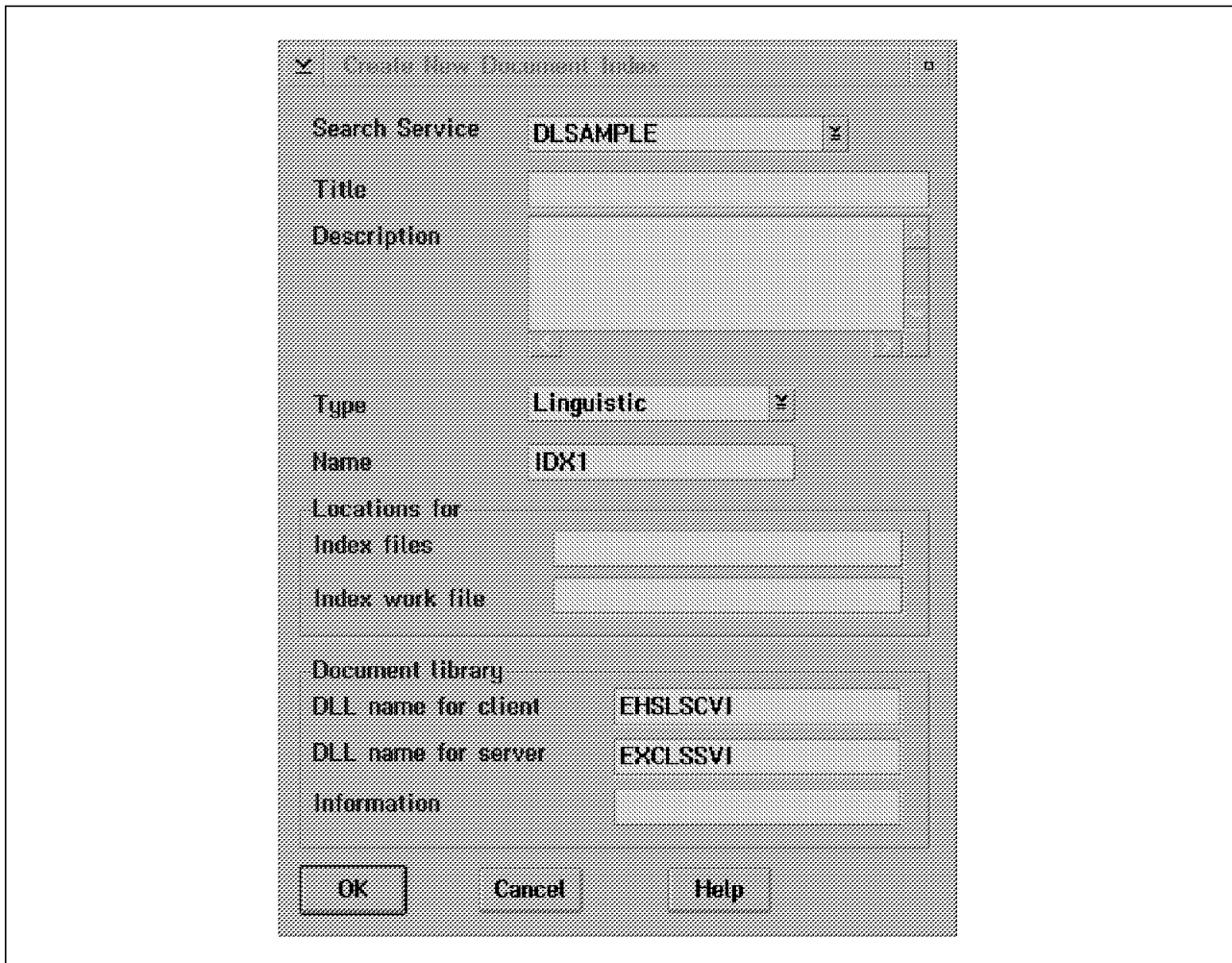
Figure 19. New Document Index

7. Creating Keys

a. In the Digital Library folder, double-click on the **Fileroom** icon. Double-click on the **Key Fields** icon. For each key field, carry out the following steps:

b. Double-click on the **New Key Field** (see Figure 20 on page 23).
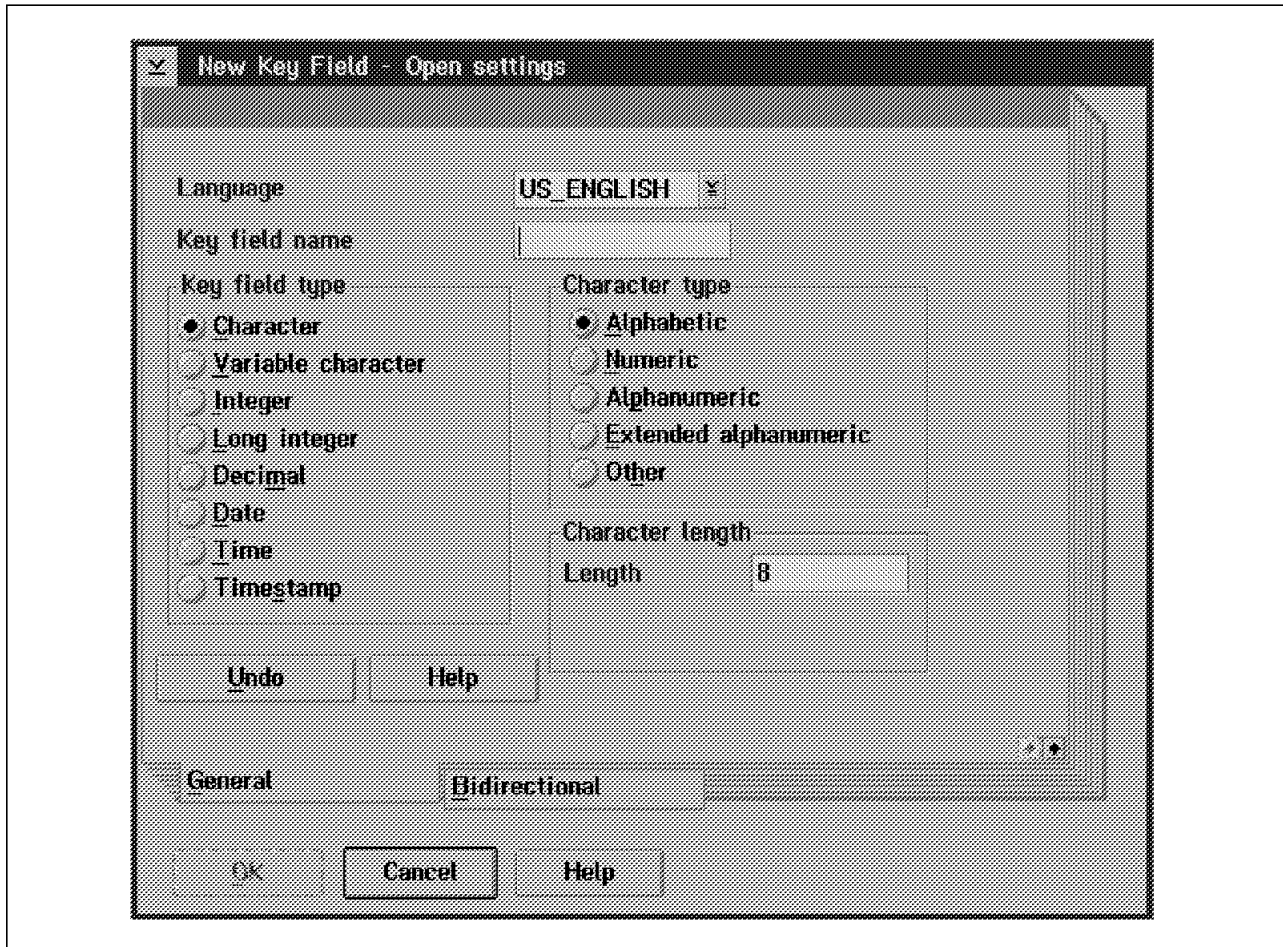
Figure 20. New Key Field

c. Enter the key field name.

d. Click on the **Variable character** radio button.

e. Enter the minimum and maximum length values.

f. Click on **OK**.

g. Close the **Key Fields** window. The names of the key fields and their respective maximum lengths are outlined in Table 4. You need to enter all of these key fields:

Table 4 (Page 1 of 2). Key Field Entries	
Key Field Name	**Max Length**
DLSEARCH_DocType	32
DLSEARCH_Date	50
DLSEARCH_Title	100
DLSEARCH_Subject	100
DLSEARCH_Author	50
DLSEARCH_Authors	100

Table 4 (Page 2 of 2). Key Field Entries	
Key Field Name	**Max Length**
DLSEARCH_Editors	50
DLSEARCH_Publisher	50
DLSEARCH_Source	100
DLSEARCH_Volume	20
DLSEARCH_Issue	20
DLSEARCH_Relation	64
DLSEARCH_DocID	32
DLSEARCH_Journal	50
DLSEARCH_Year	20

8. Creating Classes

   a. Double-click on the **New Index Class** icon. In the **New Index Class—Open settings** windows, create the following two index classes for the sample data as shown in Figure 21 on page 25:

   - DLSEARCH_MAGAZINES
   - DLSEARCH_JOURNALS
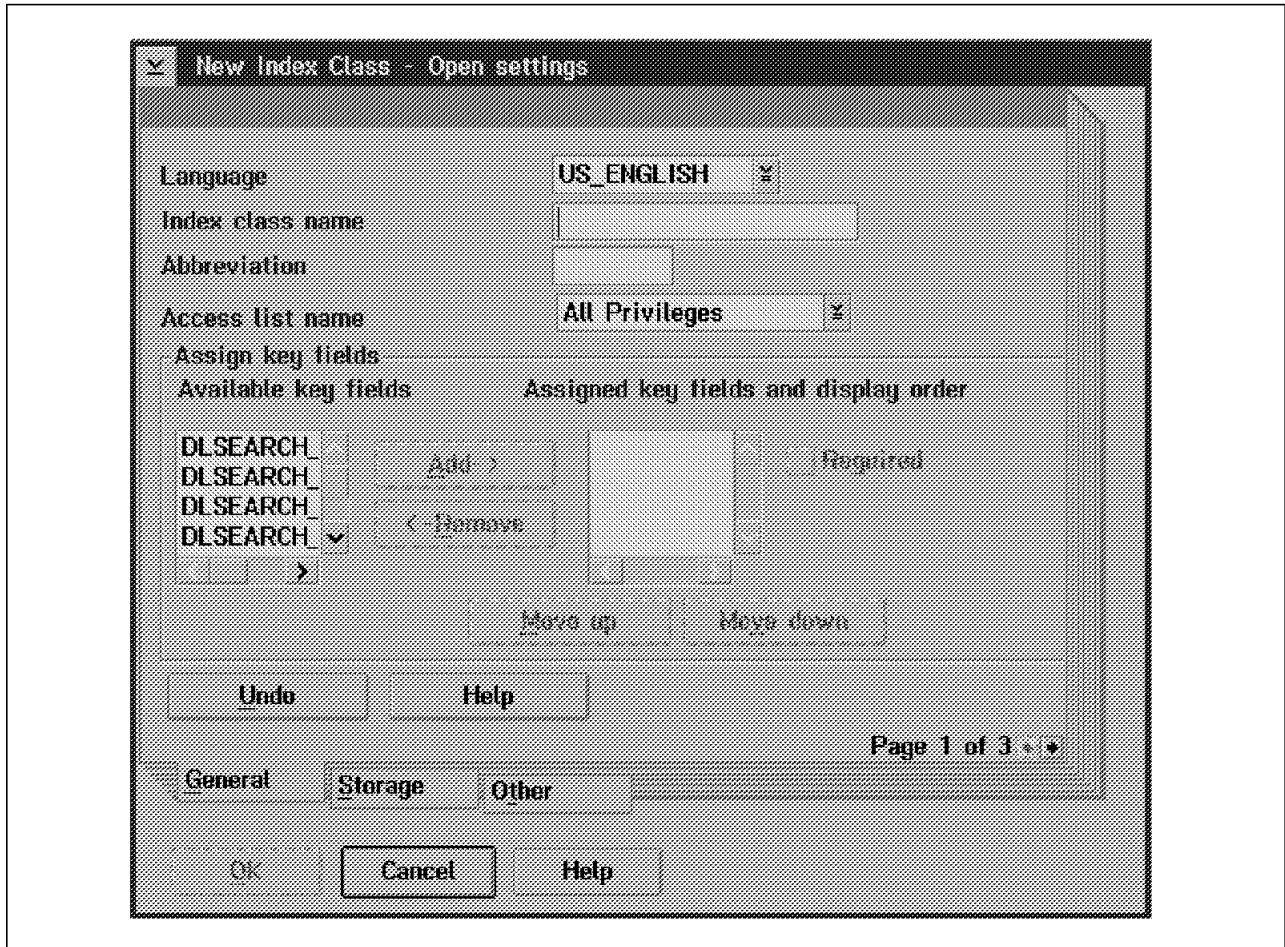
Figure 21. New Index Class

b. For each Index Class above, select the key fields from the **Available Key Fields** scrolling window in the order listed below. For DLSEARCH_MAGAZINES, the fields are:

- DLSEARCH_DocType
- DLSEARCH_Date
- DLSEARCH_Title (mark this field **Required**)
- DLSEARCH_Subject
- DLSEARCH_Author
- DLSEARCH_Authors
- DLSEARCH_Editors
- DLSEARCH_Publisher
- DLSEARCH_Source
- DLSEARCH_Volume
- DLSEARCH_Issue
- DLSEARCH_Relation

c. For DLSEARCH_JOURNALS, the fields are:

- DLSEARCH_DocID
- DLSEARCH_DocType
- DLSEARCH_Journal
- DLSEARCH_Volume
- DLSEARCH_Issue
- DLSEARCH_Year

- DLSEARCH_Title (mark this field **Required**)
- DLSEARCH_Author

    d. For each of the above index classes, click on the **Storage** button, click on the down arrow and select **OBJSRVRX** server for this index class. Click on the down arrow and select a collection, such as **CBR.CLLCT001**. Then click on **OK**. The new index class is created and available.

2.6 Loading Sample Data

Ensure that the Library Server, the Object Server, and SM6000 are all running before loading data. Follow these steps for loading the sample data:

1. Load Sample1 on the AIX that is running the Library Server (libsrvr). See 3.2.4, "Loading Sample Data" on page 38.

2. Double-click on the **System Administration** icon.

3. Double-click first on **Text Search Administration** and then on **Index Administration**. There should be 82 items in the queue, waiting to be indexed.

4. Click on **Index**, wait for roughly 3 to 5 minutes, then click on **Refresh**. The queue should now contain zero items, and the Primary Index should now contain the 82 indexed items. There are no visible indicators to signal you to press the refresh button, but waiting about 5 minutes should be enough.

5. Load Sample2 on the AIX that is running the Library Server. See 3.2.4, "Loading Sample Data" on page 38.

6. Double-click on the **System Administration** icon.

7. Double-click first on **Text Search Administration** and then on **Index Administration**. There should be 19 items in the queue, waiting to be indexed.

8. Click on **Index**, wait for approximately 3 to 5 minutes, then click on **Refresh**. The queue should now contain zero items, and the Primary Index should now contain 82 indexed items and the secondary queue should contain 19 items.

2.7 Testing Sample System

In this section, we explain how to run the Sample System that comes with IBM Digital Library 1.0. This will ensure that Object Server, Library Server, and Cliette are working properly. Make sure that you have access to a Web browser. We use a Netscape browser running on OS/2 on the machine that runs the DL Admin. The steps are these:

1. Start Cache Manager and Connection Manager on AIX to view the sample data (refer to 3.2.5, "Viewing Sample Data" on page 38).

2. Start Netscape and go to the Web page

   `http://<ip_address>/dlsearch/dlsearch.html`

   You should now get a Web page as shown in Figure 22 on page 27.
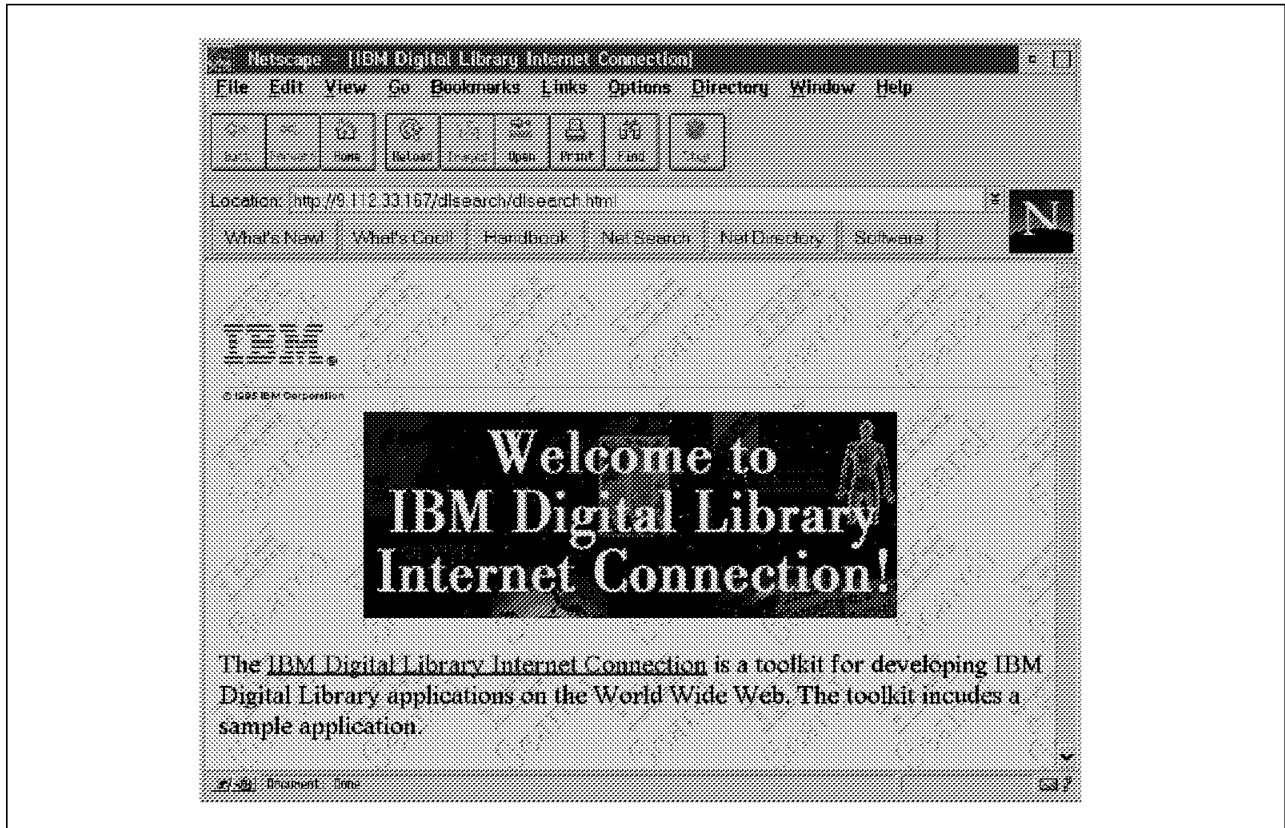
Figure 22. Digital Library—Sample System

3. Click the **start** button and you get another Web page (Figure 23 on page 28).
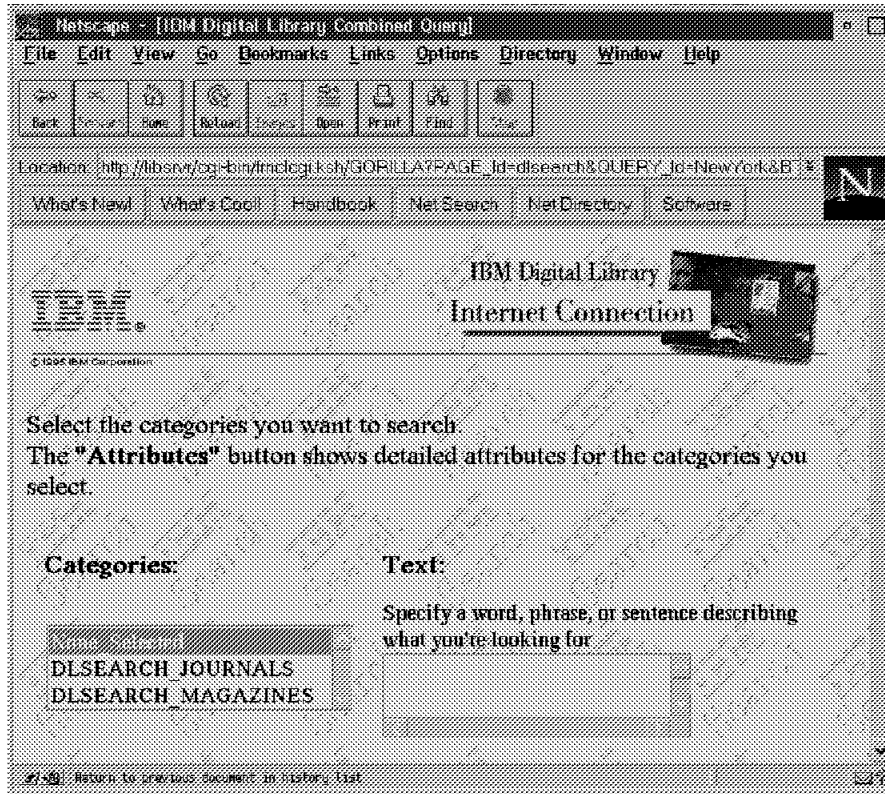
Figure 23. Sample System—Search

4. Select a search parameter. We select DLSEARCH_JOURNALS and enter "IBM" as the keyword for an optional Text search.

5. Parametric searches can also be used by setting values to attributes in the Attributes list. Continue testing by experimenting with various search parameters.

Chapter 3. Library Server and Cliette

This chapter describes in detail the installation of the Library Server, the Text Search Server (Search Manager), the Internet Connection Server for AIX (Cliette) and the configuration of these for loading and viewing the sample data. The main points are these:

- Library Server catalogs all the objects that are stored in the Digital Library System. These objects can include images, graphics, audio, video, text, and spread sheets. In addition, the Library Server also stores workflow definitions used by folders, maintains index information, and controls access for objects that are stored in the Object Server. More specifically, the Library Server directs requests from the clients to the appropriate Object Server to perform the following functions:

  - Store, retrieve, and update objects in the Object Servers.
  - Update and query the indices and other information stored in the library catalog.

- 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.

- The Internet Connection performs accessing of data stored in the Object Server through a Web interface.

Prerequisites

Before you start the installation, make sure that all the required software and hardware is available for the complete functioning of the Library Server and Cliette. The following are the requirements:

- AIX Version 4.1.5.
- C Set ++ for AIX 3.1.3. For instructions on how to install using SMIT, see A.5, "Installation of Any Software" on page 84
- Set the Language Environment of AIX to IBM-850. Refer to A.2, "Setting Language Environment to IBM-850." on page 77 on how to set the language environment.
- DB2/6000 for AIX 2.1.2. See Appendix B, "Installation of DB2/6000" on page 87.
- PERL and IBM HTTP server 4.1. See 3.2.1, "Installing PERL and HTTP Software" on page 34.

In addition, complete the following:

1. Create filesystems /libsrvr and /portal, each with 200 MB.
2. Create administrative groups dba and smadmin.
3. Create users libusr (home directory: /libsrvr,group: dba) and portal (home directory: /portal,group: smadmin).
4. Add the groups *system,dba,smadmin* to the group set of these users, and to the *root* user as well.

For more information on how to create users and filesystems, refer to A.1, "Usage of SMIT" on page 77.

3.1 Installation and Configuration of the Library Server

This section gives the detailed steps for installing the Library Server on AIX.

3.1.1 Installing the Library Server

1. Put the Digital Library CDROM in the drive.
2. Log on as the *root* user and go to the **smit** screen. See A.5, "Installation of Any Software" on page 84 for more details on using **smit**.
3. Select the device or directory from the list of available sources; for example, /dev/cd0.
4. Select the following options (see Figure 24):

   1.1.0 Library Server for AIX using DB2

5. Click the **OK** button to start the installation program. (This automatically selects the **Common Utilities for AIX**.)
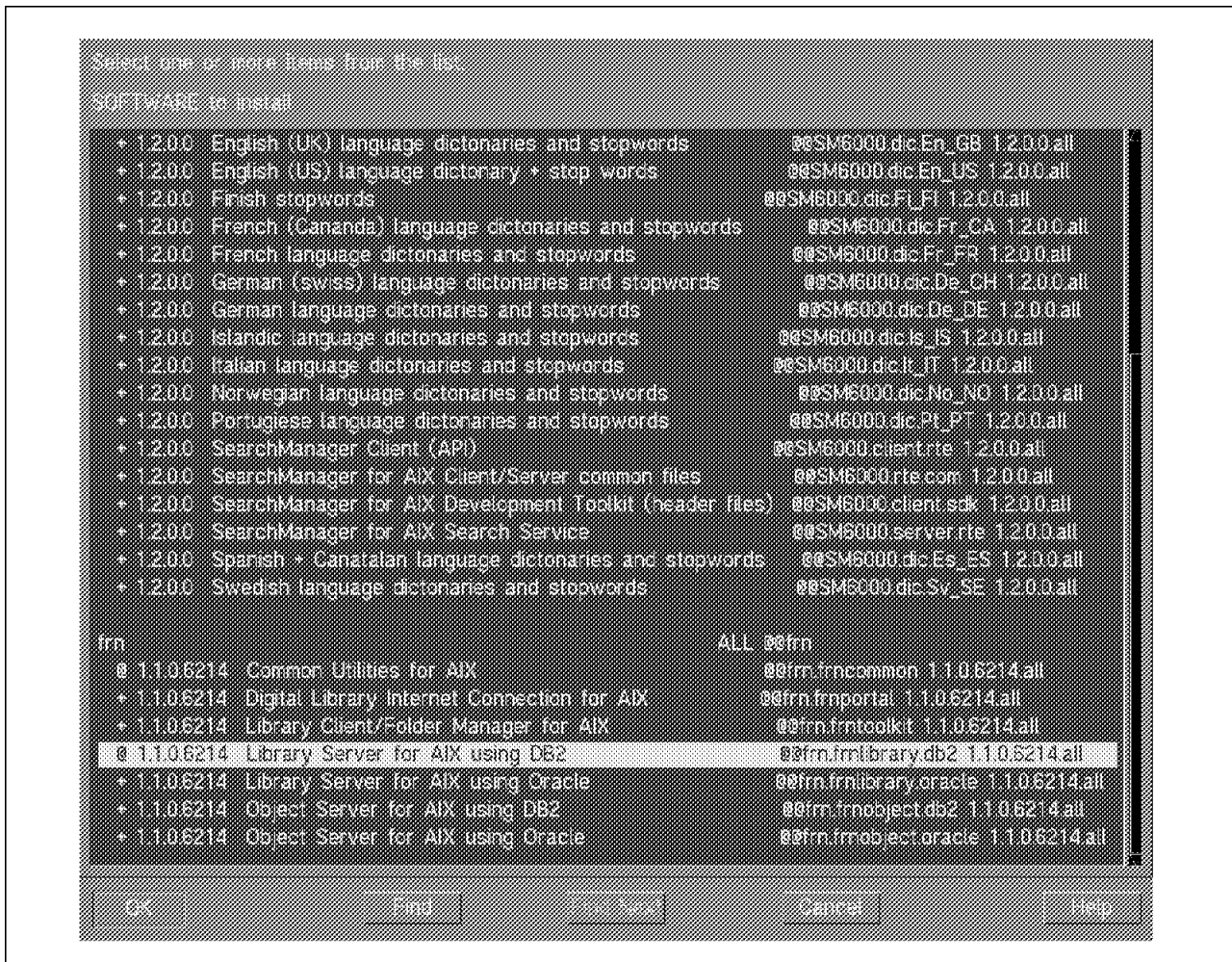
Figure 24. Library Server Installation. Installation of Library Server using smit.

Digital Library 1.0 requires that fix UR46674 be installed so that the Library Server works properly. This fix is available from the site *ftp://service.boulder.ibm.com/ps /products/digital_library/fixes/v1.0/enu/csd1/aix/*. Download this fix and install it through **smit**. (For details refer to A.6, "How to Install Fixes" on page 85.)

NOTE: You may have to reboot the machine after installing a fix.

3.1.2 Generating the Library Server Setup File

The setup file contains all the environment variables for running the Library Server. This file has to be executed before running any commands or scripts as a user of Library Server. Alternatively, you may add a line to execute the setup file in the *.profile* of the Library Server user (**libusr**).

For each separate instance of the Library Server, follow these steps to generate a setup file:

1. Log on to the Library User ID (for example, su - libusr)

2. Change to the following directory:

        $ cd /usr/lpp/frn/bin

3. Run the following:

        $ ./frnconfig.lib

You are prompted for a library server database name. Enter LIBSRVRX. The setup script is generated and placed in the following directory:

        $HOME/<prodname>/LIBSERV/LIBSRVRX/CTL/frnsetup.lib
        (for example : /libsrvr/frn/LIBSERV/LIBSRVRX/CTL/frnsetup.lib)

This setup file has to be run in the library user shell before executing any other script. An easy way would be to enter the following line in the *.profile* of the library server user:

        . $HOME/<prodname>/LIBSERV/LIBSRVRX/CTL/frnsetup.lib
        (eg: . /libsrvr/frn/LIBSERV/LIBSRVRX/CTL/frnsetup.lib)

This executes the *frnsetup.lib* file every time you log on as the library server user.

3.1.3 Generating a Network Table File

Digital Library uses the network table file to communicate between components. A sample network table file is installed by default in *$FRNCTL.* You need to generate a custom network table file that contains information about the specific setup.

To generate a network table file:

1. Log on to the system using the *root* user ID

2. Change directory to the following.

        # cd /usr/lpp/frn/bin

3. Run the following command:

        # ./frntable

4. Specify the configuration file when prompted for it (for example, */tmp/myconfig.cfg,*). For details, refer to 2.2, "Digital Library System Configuration" on page 12. Then enter the node name (or nickname) for the Library Server machine as given in the configuration file.

The network table will be named FRNOLINT.TBL and will be placed in the $FRNSHARED directory. It returns a message ″THE UTILITY SUCCESSFULLY CREATED THE NETWORK TABLE.″

3.1.4 Generating the Library Server Database

We install the database on DB/2 6000. To create a Library Server Database, do the following:

1. Log on to your library server as user (for example, su - libusr).

2. Make sure you have a DB2 instance for this user (for more information refer to Appendix B, "Installation of DB2/6000" on page 87).

3. Ensure that the setup file *(frnsetup.lib)* is executed.

4. Change the directory to

       ``` 
       $ cd /usr/lpp/frn/bin
       ```

5. Run the following;

       ```
       $ frncrtdb.lib
       ```

Name the object server database as OBJSRVRX and enter default for others. This creates the Library Server Database and returns a "successfully created" message. It then creates all the tables and indexes.

3.1.5 Starting the Library Server

Carry out these steps:

1. Log on as library server user.

2. Ensure that the setup file is executed.

3. Enter the following command

       ```
       $ frnstart.lib
       ```

When the Library Server runs for the first time, it builds the shared libraries and indicates the optimization status. The last shared library to be generated is *FRNA0080.* Refer to Figure 25 on page 33.

Figure 25. Starting Library Server. Building Shared Libraries and Optimization Status

When the Library Server is in full operation it shows two messages;
"Initializing..." and "Accepting Requests..."

3.1.6 Stopping the Library Server

When the Library Server is being stopped for the first time, begin with Step 1;
otherwise, start from Step 5.

1. Log on as *root*.

2. Change the directory to

    $ cd /usr/lpp/frn/bin

3. Edit the *frnstop* file and change all instances of "user" to "ruser."

4. Save and quit.

5. Log on as Library Server user.

6. Ensure that the environment variables setup file is executed.

7. Run the following command:

    $ frnstop

8. This shuts down the Library Server.

3.1.6.1 Statistical Information about the Library Server

You can obtain the status information and generate statistics on the current
instance of the Library Sever by entering the following command after logging in
as Library Server user.

    $ frnstats.lib

A file named $HOME/frn/LIBSERV/LIBSRVRX/LOG/LIBSRVRX.USE is generated.

3.2 Installation and Configuration of Cliette

Cliette is the Internet Connection segment of IBM Digital Library. The Digital Library Internet Connection is an application development toolkit that extends the Digital Library client for AIX. It lets you build query applications for the World Wide Web. The internet connection also provides a complete sample application designed to run from a Web browser.

The Cliette (alternatively called *Portal*) has the components Text Search Server (Search Manager), Connection Manager (CM), Cache Manager, End User Interface Services (EUIS) and Page Builder.

The Search Manager (SM) is used for fast text search using the keys and indexes. The Cliette is the front interface end between a Web browser and Library Server. The Connection manager and the Cache manager manage the communication part of the Cliette and EUIS/Page builder manages the user interface portion of the data.

3.2.1 Installing PERL and HTTP Software

The Cliette requires PERL and HTTP software to be installed as prerequisites.

PERL V 5.002 is available from the AIX public domain shareware library at *http://aixpdslib.seas.ucla.edu/languages/perl4.html.* Download the PERL software and copy it in the root directory. Untar it to have it installed in */usr/local/bin/perl.* (Add */usr/local/bin* in the root PATH if needed.) To verify whether the PERL program is working, do the following.

1. From the root prompt, enter:

        # which perl

    It should return:

        # /usr/local/bin/perl

2. To check further, do the following:

        # perl

    This should return with no command prompt. Enter:

        print "Hello World\n";

    Press a *control-D.* It should answer as

        # Hello World

An HTTP server is required to be installed in the portal machine. The IBM Internet Connection Server 4.1 (HTTP server) is available from *http://ics.raleigh.ibm.com/ics/index.html* . Unpack this into */usr/sys/inst.images* directory and run **smit** to install this server. Refer to A.5, "Installation of Any Software" on page 84 for details of using smit.

3.2.2 Installing and Configuring Search Manager

The Search Manager is installed on the Portal (Cliette) machine. In our case, the Portal is installed on the same machine as the Library Server. Follow the steps below for installing and configuring the Search Manager:

Insert the Digital Library software CD and install the following using smit (see Figure 26 on page 35 and for details on smit see A.5, "Installation of Any Software" on page 84):

SM Client
SM Client/Server common files
SM Development Toolkit,
SM for AIX search service
Digital Library Internet Connection for AIX
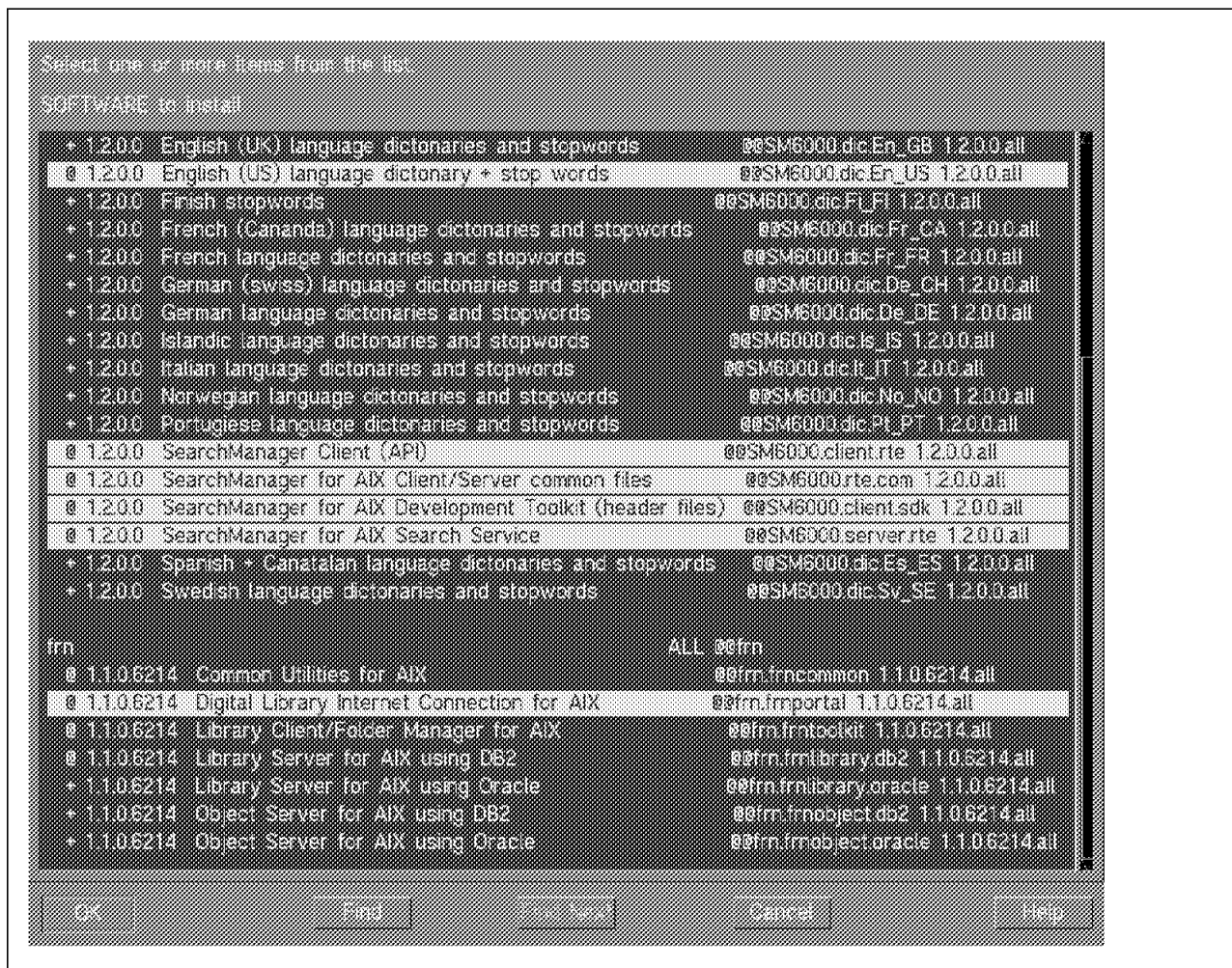
Figure 26. The Search Manager Installation Using SMIT

Search Manager also requires a fix UR47604 to be installed. This is available at the site *http://www.service.boulder.ibm.com/ ps/products/digital_library/fixes/v1.0/enu/csd2/aix*. Install this fix, also using SMIT, after selecting all the appropriate components as given below:

- **1.2.0.0 English (US) language dictionary+stop words**
- **1.2.1.0 Text Search Server Client (API)**
- **1.2.1.0 Text Search Server for AIX Client/Server Common files**
- **1.2.1.0 Text Search Server for AIX Development Toolkit**
- **1.2.1.0 Text Search Server for AIX Search Service**

There is an additional fix for cliette to be installed. This is available at the site *ftp.stl.ibm.com*. Log on as *anonymous* and give your e-mail ID (or any e-mail ID)

as password. Change directories to *dl/dlxnls15*. Download the following from this directory and install using **smit**:

- README.AIX.ENU
- SM6000.pkg
- README.SM
- frnportalcom.img.7066
- frnportalenus.img.7066

NOTE: The machine needs to be rebooted after installing these patches.

Next, remaining as *root* user, change directory to */usr/lpp/SM6000* and create the following directories.

- client
- client/index
- client/work
- index
- index/dlsearch
- work
- work/dlsearch

Change ownership of all these directories to Portal as follows.

```
chown -R portal.smadmin client index work
```

Give execute permission to the *dlsearch* directories. Log on as *portal*, and in the *.profile* insert the following environment variables:

```
export EXCDATASRV=/usr/lpp/SM6000/index
export EXCWORKSRV=/usr/lpp/SM6000/work
export EXCNLPSSRV=/usr/lpp/frn/SM6000/dicts
export EXCDATACL=/usr/lpp/SM6000/client/index
export EXCWORKCL=/usr/lpp/SM6000/client/work
export EXCNLPSCL=/usr/lpp/SM6000/dicts
export EXCINSTENV=DLSAMPLE
```

Execute *.profile*

To configure the Search Manager for the instance DLSAMPLE, log on as *portal* and enter the following commands.

```
$ smconfig -create -server -in DLSAMPLE -sn DLSAMPLE -mi /usr/lpp/
SM6000/index  -mw /usr/lpp/SM6000/work -ip nnn.nnn.nnn.nnn -sp 7502
```

```
$ smconfig -create -client -in DLSAMPLE -sn DLSAMPLE -ip nnn.nnn.nnn.nnn -sp 7502.
```

where nnn.nnn.nnn.nnn is the IP address of the machine where Portal is installed. The number 7502 is the port number. Do not change it. Leave it at 7502. Successful execution will return the message REQUEST COMPLETED SUCCESSFULLY.

Log out of the Portal and, as *root* user, execute an **xhost +** command at the prompt. (This is to indicate that any client can log on from any host.) Log on as portal once again. Do an **export DISPLAY=<hostname>:0** command at the prompt. Change directory to /usr/lpp/SM6000/index. Run command **excdlpwu**. This will open a window and ask for a user ID and password. Enter FRNADMIN for user ID and PASSWORD for password. The password will be shown on the screen. Click **OK**. This creates a file called *.ehsvipwf.dat.* The size of this file should be 284 bytes. If not, repeat the command **excdlpwu** once again.

Next go to the HOME directory of *portal.* Add the following line to the *.profile* of the *portal* and execute the *.profile.*

```
EXPORT PORTAL=/usr/lpp/frn/samples/cliette
```

Change directory to $PORTAL. Ensure that it is in /usr/lpp/frn/samples/cliette directory. Run **newhost.post_i** script. Successful completion will return messages that two files *frncliet.ini* and *dlsearch.html* have been updated. Verify that these files contain the IP address and host name of your machine where *portal* is installed.

NOTE: You may also need to change the library server database name to LIBSRVRX manually in these files.

Change directory to /usr/lpp/frn/bin and run the file *frnconfig.clt.* You will be prompted for a variety of inputs. Take all the defaults. Successful completion of this generates another file called *frnsetup.clt*, and the final output of the file *frnconfig.clt* will ask you to run *frnsetup.clt.* This is the environment variable setup file for the Portal. This file has to be run before you execute any other scripts as user *portal.* Otherwise you could add the following line in the *.profile* of *portal* so that it's executed whenever logged in:

```
        $HOME/portal/frn/CLIENT/PORTAL/CTL/frnsetup.clt
```

Finally, you need to configure the Web server to serve up the IBM Digital Library.

3.2.2.1 Web Server Configuration

First, from root, change directory to /usr/lpp/internet/server_root/cgi_bin directory. (If you have a different http server, go to *cgi_bin* directory.) From this directory, run the following link command:

```
# ln -s /usr/lpp/frn/samples/cliette/frnclcgi.ksh
```

Verify that it is linked properly. Set up an alias to the dlsearch. The exact command varies depending on the Web server used. If you are using an *IBM Internet Connection Server*, then there is a file called *httpd.conf* in the */etc* directory. As *root* user, add the following line in this file:

```
Pass /dlsearch/*        /usr/lpp/frn/samples/cliette/*
```

This alias statement translates all URL references to /dlsearch/ into /usr/lpp/frn/samples/cliette. The Web server configuration is now complete.

Move to the OS/2 machine and follow directions as given in 2.4, "Library Server Configuration from OS/2" on page 18.

3.2.3 Starting and Stopping Search Manager

Log on as portal and change directory to $PORTAL. Make sure that the setup file is executed. To start the search manager, run the following command:

```
        $ sm6000 start DLSAMPLE DLSAMPLE
```

When you give the command **ps -fuportal** you should be able to see some DLSAMPLE processes running. You could also check the search manager configuration status command as follows.

```
        $ sm6000 status DLSAMPLE
```

To stop the search manager, give the following command.

```
        $ sm6000 stop DLSAMPLE DLSAMPLE
```

To increase the timeout limit of the Search Manager server, give the following command after ensuring that the SM6000 server and cache manager are shut down:

```
$ smconfig -update -server -in DLSAMPLE -sn DLSAMPLE -tl <time in seconds>
```

Restart SM6000 and the Cache Manager after this completes.

3.2.4 Loading Sample Data

The sample data consists of articles taken from three IBM magazines: articles from *Software Quarterly* are coded in HTML and embed image files; articles from *The IBM Journal of Research and Development* and *The IBM Systems Journal* are flat text files. Internet Connection provides a sample program that loads the articles into a Digital Library Object Server.

3.2.4.1 Steps for Loading Sample Data

Carry out these steps:

1. Start Object Server and Library Server.

2. Start Search Manager.

3. Log on as portal. Change directories to /usr/lpp/frn/samples/osqsdata. Check that the data-loading scripts, *ldsample1* and *ldsample2*, are in this directory. You may need to open these script files using *vi* and change all the instances of LIBSRVR2 to LIBSRVRX.

4. Change to the subdirectory /usr/lpp/frn/samples/oqsdata/magazine. This contains a file *sample1.tar.* Untar this file by issuing a **tar xvf sample1.tar** command.

5. Change directories to /usr/lpp/frn/samples/oqsdata/journals and check if it contains the file *sample2.tar.* Untar this file using **tar xvf sample2.tar** command.

6. Change directories back to /usr/lpp/frn/samples/oqsdata and run *ldsample1* program. It should run for approximately 30 minutes.

7. Move to the OS/2 machine and follow the procedure described in 2.6, "Loading Sample Data" on page 26.

8. Ensure that you are still in /usr/lpp/frn/samples/oqsdata. Run the program *ldsample2.* This should run for about 10 minutes.

9. Once again, move to the OS/2 machine and follow the procedure given in Section 2.6, "Loading Sample Data" on page 26.

3.2.5 Viewing Sample Data

The sample data loaded can now be viewed through a normal Web browser.

3.2.5.1 Steps for Viewing Sample Data

Follow these steps:

1. Log on as portal and change directories to $PORTAL.

2. Start SM6000 server.

3. Run the following two commands (in order):

```
$ cachemgrd -c frncache.cfg &
```

This command should display a process ID and return you a new command prompt. Next run:

```
$ CMDaemon -p 7130 frncmmgr.cfg
```

This command should return the message: CLIETTE IS READY.

4. You are now able to view the sample data by directing a Web browser to http://*hostname/dlsearch/dlsearch.html.* For more details, see 2.7, "Testing Sample System" on page 26.

3.3 Troubleshooting Tips

If the Client or DL Admin fails to connect to the Library Server, check:

1. TCP/IP connection.
2. Whether the Primary Language Environment is IBM-850 English (United States).
3. Whether the network table has been created properly and it's in $FRNCTL or $FRNSHARED directory.
4. Whether the Library Server Database has been bound properly.

If the Library server does not start, check:

1. For the library server database in the DB/2.
2. Whether patches have been installed and machine has been rebooted.

If the sm6000 server processes do not run or start:

After you start the SM server, give the command **ps -fuportal** and, if you find that the sm6000 processes are not running, check whether fix U47604 has been installed properly. This fix has to be installed for proper running of the search manager server in IBM Digital Library 1.0. The machine must be rebooted after the installation of the patches.

If the data loading scripts does not run within the specified time.

If the data loading scripts *ldsample1* and *ldsample2* do not run for the specified time, then the data may not be getting loaded properly. Check whether the Object Server is running. Also the *Requests Processed* field in the Object Server screen gets updated when this script runs.

If the Cliette returns an error message in the Digital Library Web page when clicking 'Start':

Check whether the portal could connect to the Library Server. An easy method could be as follows.

Log on as *portal.* Change directory to /usr/lpp/frn/samples/oqs. Run the script *frnsampq.* It gives a list of options. Select option **1**, which asks for the VisualInfo library specification. Enter the command given in the example with **LIBSRVR2** (replaced by **LIBSRVRX** in our application). Press Enter.

The portal gives a message whether it connects to the Library Server or not. If it still does not connect, there is something wrong in the portal installation. Retrace the installation steps and check for errors.

If the Cliette gives an error message while a search query is posted:

1. Check whether the search manager and cache manager are running.

2. Check whether the time-out limit of the search manager has been reached. If yes, then stop the search manager and the cache manager, increase the time limit and then restart the search manager and the cache manager. (Refer to 3.2.3, "Starting and Stopping Search Manager" on page 37.)

3. Check whether the portal fix (7066) has been installed properly. (Refer to 3.2.2, "Installing and Configuring Search Manager" on page 34.)

Chapter 4. Object Server for AIX

This chapter describes how to install/configure the Object Server in AIX for Digital Library Version 1.0.

4.1 Install Object Server for AIX

The object server manages multimedia objects which you store in the Digital Library. This section describes how to install the Object Server for AIX.

4.1.1 Prerequisites

Before you start the installation, make sure that all the prerequisites are available for the complete functioning of the Object Server. The requirements are these:

- AIX version 4.1.5.
- C for AIX 3.1.3.
- C Set ++ for AIX 3.1.3.
    (For instructions on how to install using smit see A.5, "Installation of Any Software" on page 84.)
- DB2/6000 for AIX 2.1.2 (See Appendix B, "Installation of DB2/6000" on page 87)

In addition, complete the following:

1. Set Language Environment to IBM-850
2. Create file systems /objsrvr, /ubosstg, and /ubosdata, each of 200 MB.
3. Create administrative groups **dba**. Add this group to the group set of the *root* user.
4. Create user *objusr(home directory: /objsrvr)*.
5. Add the group *system* to the group set of this user.

For more information on how to create users and file systems, see A.1, "Usage of SMIT" on page 77.

4.1.2 Installing Object Server Code

Carry out these steps:

1. Log on as user root.

2. Type smit at the command prompt.

3. Open **Software Installation and Maintenance** from the list of choices on the System Management Panel.

4. Select **Install and Update Software**.

5. Select **Install/Update Selectable Software (Custom Install)**.

6. Select **Install Software Products at Latest Level**.

7. Select **Install New Software Products at Latest Level**.

8. Click **List** to select the device or directory from the list of available sources, or specify the input device or directory that is the source of the software you are installing. The input device can be a tape, file system, CDROM, diskette drive, or a directory.

    Then click **OK**.

9. Click on **List** of **SOFTWARE to install** to list the available software.

10. Select an object server to install, for example, Object Server for AIX using DB2.

    Then click **OK**.

    When you select a Object Server for AIX, the common utilities are installed automatically. The common utilities include the IBM C++ runtime libraries, *libC.a*, which are necessary for running Digital Library. The version of *libC.a* installed with Digital Library does not overwrite the *libC.a* in */usr/lib*, which is required in location of *libC.a* installed with C++ compiler.

11. Accept the SMIT defaults for the installation option on the **Install Software Products at Latest Level** window.

    If the current level of the object server is already installed, select **OVERWRITE same or newer version** and change *no* to *yes*, or uninstall the existing version.

    Then click **OK**.

12. Click **OK** on the confirmation dialog box to start the installation program.

13. Click **DONE** when the installation program completes successfully.

14. Close SMIT.
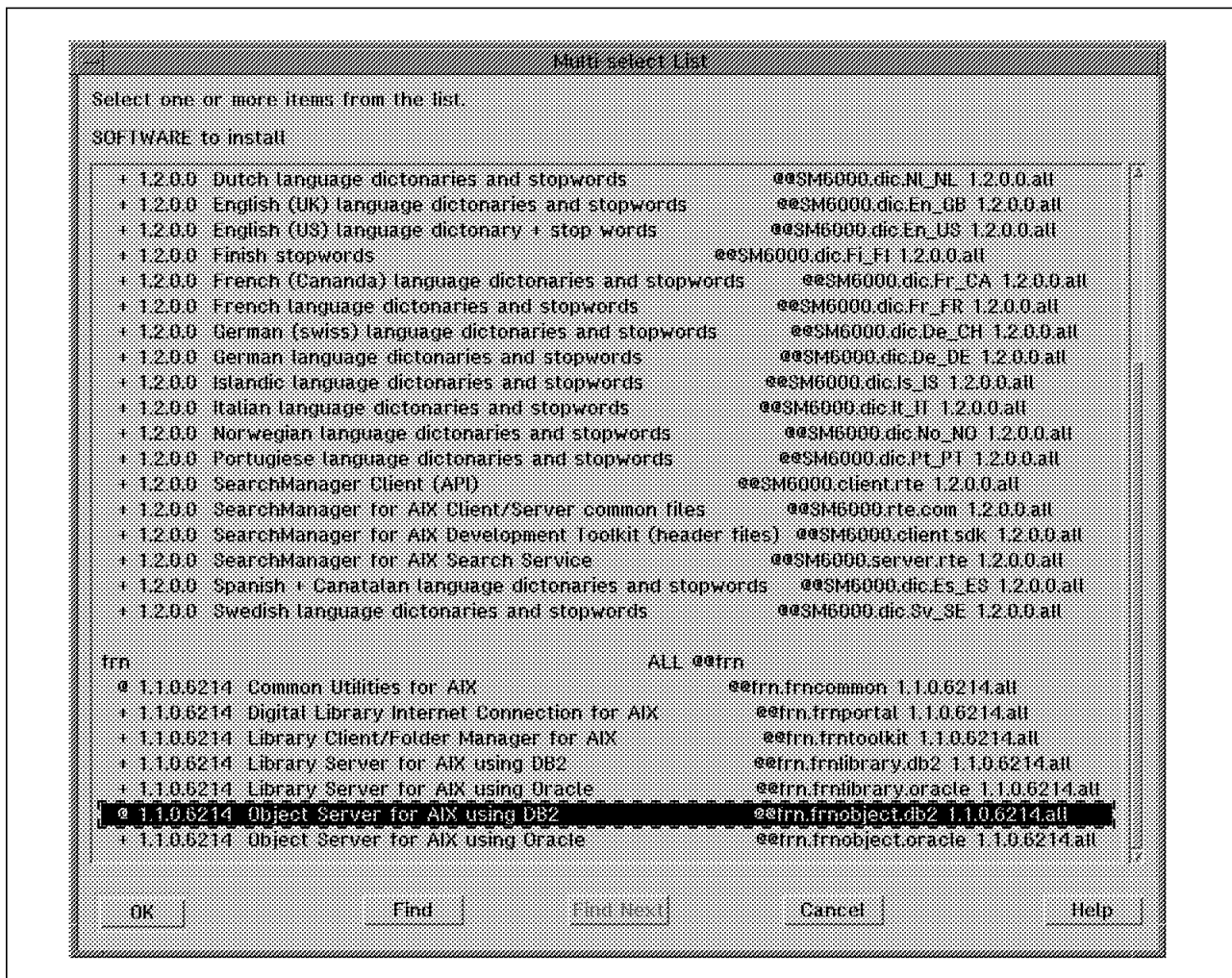
Figure 27. Install Object Server Code

4.1.3 Apply Fix (UR46674)

FixPak UR46674 is available now. Please download it from the following site:

`ftp://service.boulder.ibm.com/ps/products/digital_library/fixes/v1.0/enu/csd1/aix/`

To install the FixPak, please refer to A.5, "Installation of Any Software" on page 84.

4.2 Configure and Run Object Server for AIX

This section gives a description of how to create an Object Server database, configure it and finally how to start and stop the Object Server.

4.2.1 Generate a Setup File

The setup file for the Object Server for AIX contains settings for required environment variables. You must run the setup file before you run any application code, otherwise the Digital Library directory will not be in the path and environment variables will not be set. The setup file will be generated for each separate instance of the object server (each object server database).

Follow these steps:

1. Log on to your object server user ID.

2. Change to the Digital Library directory.

    $ cd /usr/lpp/frn/bin

3. Run **frnconfig.obj** to create the object server setup script:

    $ ./frnconfig.obj

4. You are prompted for the object server database name. You may specify a name, but during the quick start you should accept the default. After you press Enter, the setup script will be generated and placed in the following directory:

    $HOME/frn/OBJSERV/<OSDBNAME>/CTL/frnsetup.obj

    where <OSDBNAME> is the name of the object server database.

This setup file has to be run in the object user shell before executing any other script. An easy way would be to enter the following line in the *.profile* of the object server user.

        $HOME/<prodname>/OBJSERV/OBJSRVRX/CTL/frnsetup.obj
(for example: /objsrvr/frn/OBJSERV/OBJSRVRX/CTL/frnsetup.obj)

This executes the *frnsetup.obj* file every time you log on on as the object server user.

4.2.2 Creating a Database for the Object Server for AIX

The Object Server for AIX database can reside on either DB2/6000 or Oracle. This section lists instructions for DB2/6000 **only**.

The general steps of creating the DB2 database are listed below. The following sections explain each task in more detail.

To create database:

1. Verify the number of DB2/6000 license.

2. Create DB2 Instance

3. Setup DB2 required environment variables.

4. Check available space on the file system.

5. Create the database.

4.2.3 Verify the Number of DB2 for AIX Licenses

Several DB2 for AIX licenses are required to run Digital Library servers. The number of DB2 for AIX licenses should at least equal the number of dependent processes, plus two. The default number of dependent processes for Digital Library is five.

4.2.4 Create DB2 Instance

Please refer to Appendix B, "Installation of DB2/6000" on page 87.

4.2.5 Setup DB2 Required Environment Variables

Carry out these steps:

1. Log on as object user ID

2. Type the following command to create .profile for object user ID:

   $ cp $HOME/sqllib/db2profile $HOME/.profile

3. Add the following commands (Figure 28) in *$HOME/.profile*:

   DB2COMM=TCPIP

   export DB2COMM

Figure 28. Setup DB2 Required Environment Variables

4.2.6 Check Available Space on the File System

Make sure that you allocate enough disk space for the file system where your DB2INSTANCE directory and Digital Library database resides.

Check the file system using the AIX command **df** and verify that the file system is not too near its capacity. As the object server database grows, you must allocate more space for the file system associated with DB2INSTANCE.

4.2.7 Create the Database

Carry out these steps:

1. Log on to your system using your object server user ID.

2. If you have not already run the object server setup script in the current session, run the *frnsetup.obj* script to set the required environment variables:

    $. $HOME/frn/OBJSERV/<OSDBNAME>/CTL/frnsetup.obj
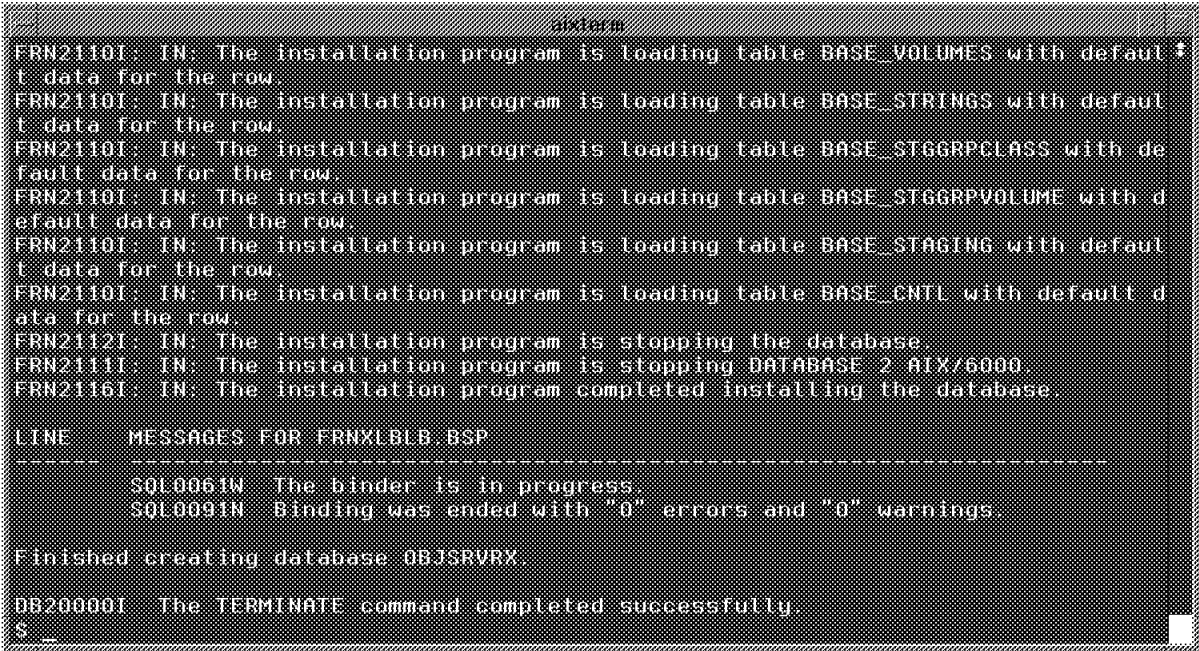
   where <OSDBNAME> is the name of the object server database.

3. Ensure you have a valid DB2INSTANCE and export it.

4. Create an object server database. Enter

    $ cd /$FRNROOT

    $ ./frncrtdb.obj

   You may specify the information for which you are prompted, but you should accept the default values in this time to set up quickly. The successful completion window is shown in Figure 29.

Figure 29. Create the Database

4.2.8 Generate Network Table

Digital Library uses network table files to communicate between components. If you are installing the object server on the same AIX machine as your library server, then your server can share one network table, and you can skip this task.

To generate the network table:
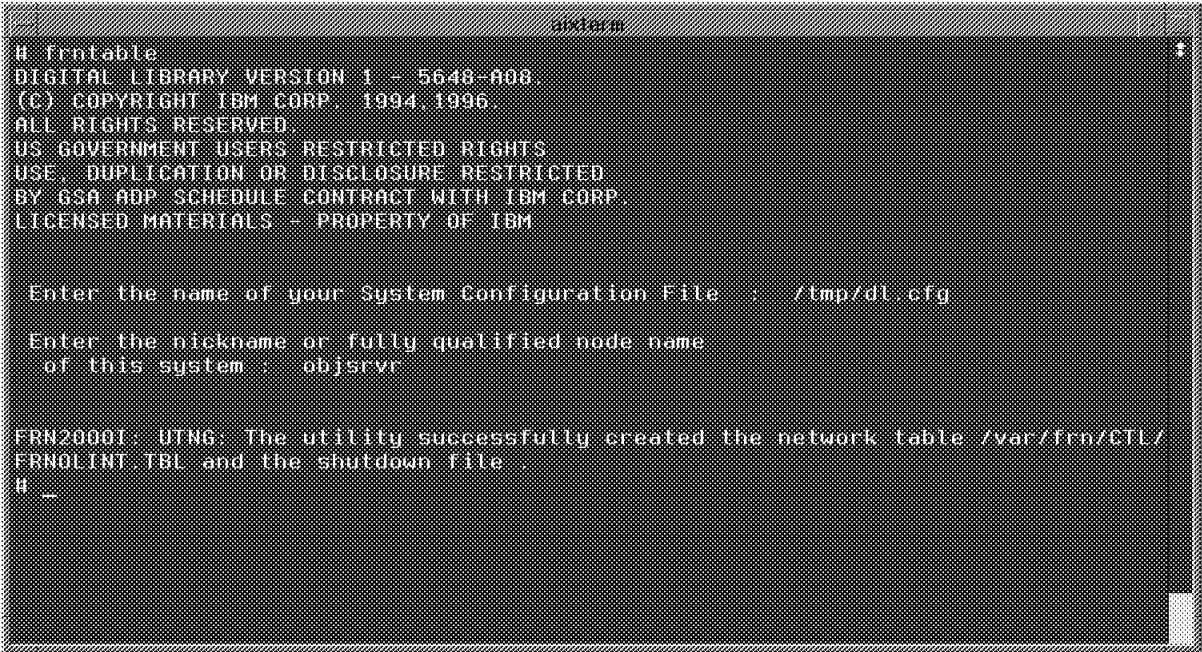
1. Log on as user *root*.

2. Transfer your system configuration file to the AIX machine on a disk or with FTP.

3. Change current directory to

   # cd /usr/lpp/frn/bin

4. To create a new network table file, enter the following command:

   # ./frntable

Specify the file name of the configuration file when prompted, for example */tmp/dl.cfg*, then enter the node or nickname that represents the Object Server for AIX. The resulting table is placed in $FRNSHARED/FRNOLINT.TBL. The successful completion window is shown in Figure 30.

Figure 30. Generate Network Table

4.2.9 Start the Object Server for AIX

> **— Note ——**
>
> You must run the Object Server for AIX under X windows. It cannot be run under CDE.

In AIX 4.1.5, to log on without entering CDE, select the Options menu on the login window, and then select **Command Line Login**. After you logged in, type `xinit` command at the command prompt to start X window, then:

1. Log on as object server user ID.

2. If you have not already run the object server setup script in the current session, run the *frnsetup.obj* script to set the required environment variables by running the following script:

   ```
   $ . $HOME/frn/OBJSERV/<OSDBNAME>/CTL/frnsetup.obj
   ```

   where <OSDBNAME> is the name of the object server database.

3. From an aixwindow, change to the Digital Library installation directory.

   ```
   $ cd $FRNROOT
   ```

4. Ensure you have a valid DB2INSTANCE and export it.

5. To start the Object Server for AIX, enter the following command:

   $ ./frnstart.obj

   The Object Server Status window opens (Figure 31).
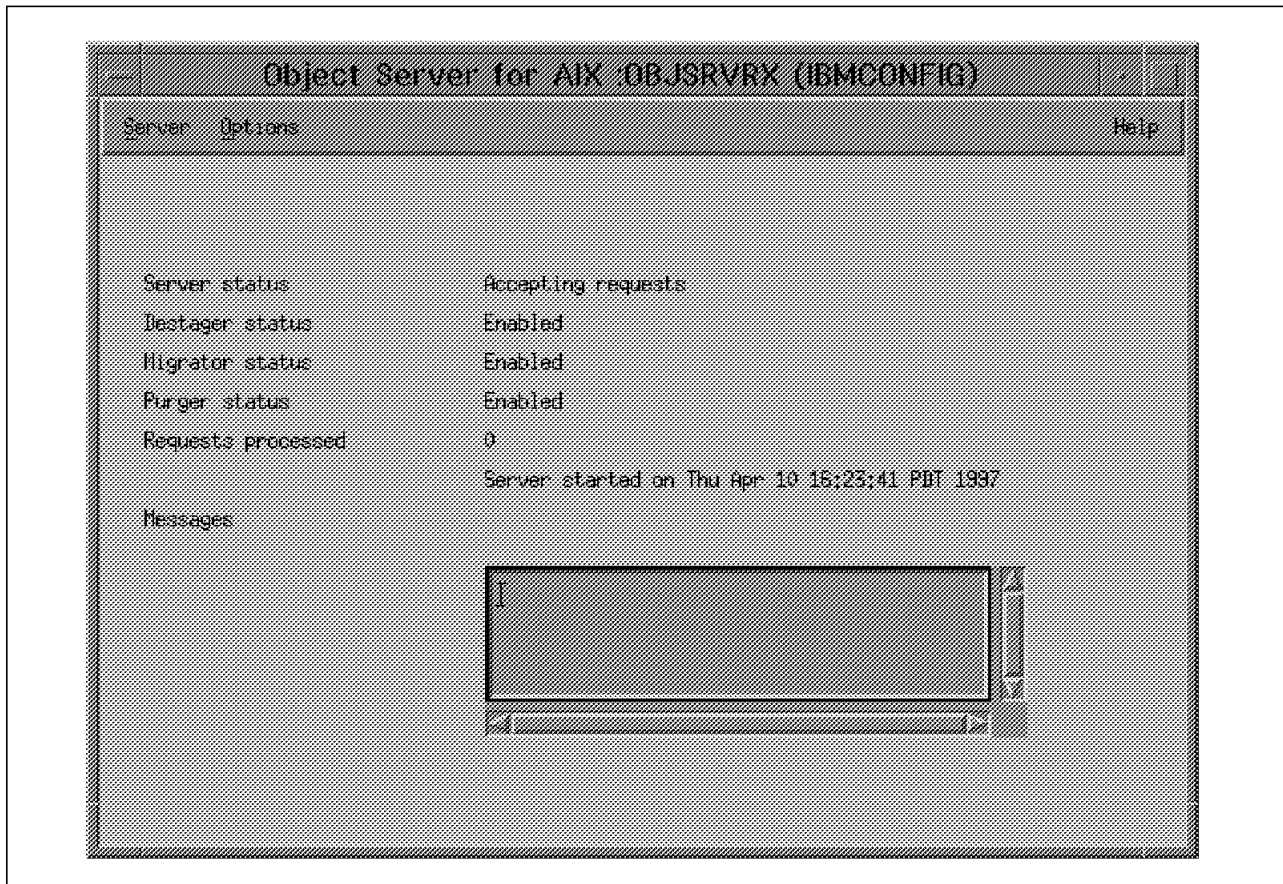
Figure 31. Start Object Server

4.2.10 Stop the Object Server for AIX

Double-click the control button in the upper left corner of Object Server for AIX window (Figure 31 on page 47). This closes the window and stops the object server.

If double-clicking the control button does not stop the Object Server for AIX, enter `./frnstop` under $FRNROOT. You may then need to run the object server recovery utility to ensure the integrity of your object server database. To run recovery utility, enter `./frnrecdb.obj` under $FRNROOT.

Chapter 5. Digital Library Client for Windows

This chapter describes an installation using the Client for Windows program of the IBM Digital Library. Our sample system is based on the client/server configuration in which a Library Server, an Object Server, a DL Admin and a Client for windows are connected by a LAN. Chapter 1, "IBM Digital Library Overview" on page 1 shows the configuration of the sample system. The client applications of the IBM Digital Library 1.0 also support a multiuser environment and accommodate multiple client platforms running OS/2, Windows, AIX, or IRIX.

The client installation described in this chapter is confined to the client for Windows 95 with the VisualInfo solution. Therefore, the "Client for Windows" in this chapter implies the VisualInfo client for Windows 95 users.

The VisualInfo client for Windows provides a user interface that enables you to bring documents into your system, then view, process, store, and retrieve them. The major Digital Library client functions are to create and capture multimedia documents, index and search, and view and process them according to the user's requests.

The VisualInfo system provides users with various ways of customizing the user environment to fit their business needs. The user's process is therefore based on the implementation and the configuration of the system your enterprise has chosen. The readers of this book need to keep in mind that the system configuration and the given user privileges determine the options and functions available for them. Your Digital Library system administrator can tell you what privileges, options and functions are available (refer back to DL Admin in Chapter 2, "Digital Library Administration and Utilities on OS/2" on page 7).

This chapter is organized as follows:

- 5.1, "VisualInfo Client for Windows 95: Overview" gives an overview of the Client for Windows concepts.

- 5.2, "Digital Library Client for Windows Installation" on page 52 describes hardware/software requirements and demonstrates the installation procedure.

- 5.3, "How to Use the Utilities of VisualInfo Client for Windows" on page 60 shows some basic operations to verify the correctness of the program install.

5.1 VisualInfo Client for Windows 95: Overview

The Client for Windows described in this book makes use of the VisualInfo solution, which is designed to simplify your multimedia-intensive document processing and increase productivity. In working with data items, the operations of Client for Windows can be characterized by: (1) create and capture, (2) index and search, and (3) view and process. Let us briefly overview the concepts that are often used in these operations.

Documents are data items that can be stored, retrieved, and delivered as a unit among systems and users. A single document may contain objects in different types of media, such as text and images.

Folders are a way to group related documents and other folders together. Grouping them in folders makes it easy for users to retrieve them. In addition, folders allow users to display search results. DL Admin can define other ways to use folders in your system.

Fileroom is the place where documents and folders are stored. A fileroom is located in Object Server and can be accessed at different levels of access privilege.

A *workbasket* is a temporary holding place for documents and folders that are either in process or waiting to be processed. Items in the workbasket are ordered much like those in a queue. A workbasket definition includes the presentation, status, and security of its contents. DL Admin can define workbaskets to organize your work. There are user-defined workbaskets and system-defined workbaskets.

A *workflow* is a predefined sequence of workbaskets through which a document or folder travels while it is being processed.

5.1.1 Creating and Capturing Documents

The Client for Windows enables you to generate documents and bring them into the system by either scanning hard copies or importing files.

You can use a scanner to convert your paper documents into digital image files that you can store and process in the Client for Windows. The Client for Windows supports various settings for your scanner (see 5.2.1, "Hardware and Software Required" on page 52).

You can also import documents from different directories on your PC or LAN. Client for Windows supports a variety of data formats for importing documents, and you can define additional ones through DL Admin.

As items enter the system, the client application can automatically sort and store them into folders. Also you can process documents according to the workbaskets to which the documents are assigned, and the workflows that connect these workbaskets. Using workflows, newly scanned or imported documents start in a specific workbasket and are then indexed and routed to other workbaskets for further processing.

5.1.2 Indexing and Searching Documents

All information stored in the digital library can be located by using a catalog formed upon various key index terms. The indexing scheme of the VisualInfo client provides the catalog function through the use of index classes. Indexing documents is for properly storing and efficiently searching the data items. When you index documents and folders, you assign them values that define where and how the Client for Windows stores and searches them. To index a document, you select an index class and type in the appropriate key field values.

Index classes are like file cabinets. They represent the main categories by which documents are stored and searched. An index class is a category used to identify a group of documents and folders stored in the fileroom. Each index class contains a defined set of *key fields* that uniquely identify the document or folder when searched. Index classes and key fields can be defined for your library system by DL Admin according to how items are stored, searched, and presented for display.

IBM Digital Library Client for Windows supports a simple indexing scheme by using a catalog function based on the use of index classes and key-field specifications.

The result is that it provides a search capability; you can search the fileroom for specific documents. You can retrieve items within a single index class, or from all classes by specifying index values as search criteria.

5.1.3 Viewing and Processing Documents

Client for Windows provides various options for retrieving and processing documents and folders, using the following operations:

Checking In/Out Documents and Folders
Checking out a document or folder ensures that only one user is working with the item at a time. When a document or folder is checked out by a user, other users can view it but cannot perform any processing tasks on it.

Moving Documents and Folders to Workbaskets
A document or folder can be moved from one workbasket to another. For e xample, you may want to move a document or folder to the workbasket of your supervisor when you find a problem in it. In a workflow, you can also move it to the next workbasket in the workflow, move it to another workflow, or remove it from the workflow altogether.

Suspending Documents and Folders
A document or folder can be made unavailable to users working with system-assigned workbaskets until a specified date and time.

Displaying Documents
A viewer of the Client for Windows offers a variety of functions for displaying document images. You can use the viewer functions as document manipulation tools to make your task easier. However, you cannot delete or update the content of the document (text or image) with the viewer unless you have been given the privilege to do so. Other display tools can help you quickly move to a page in a document or group of documents, or zoom in on a portion of a document.

Annotating Documents
You can annotate documents in many different ways if the DL Admin has given you the authority to do so. For example, you may want to add highlighting or notes to an important part of a document.

Exporting Documents
You can export documents from the Client for Windows to other directories on your PC or LAN in order to use them with another application or route them to another location.

Printing Documents
You can print documents open in the viewer window. You can select a printer on your network, set paper size, and set orientation.

5.2 Digital Library Client for Windows Installation

This section presents the procedure for installing the VisualInfo Client for Windows in the IBM Digital Library Version 1.0. Before you install, make sure your machine has the correct hardware and software components. The following two subsections give the hardware and software requirements for installing the VisualInfo Client for Windows 95, and a full description of installation procedure, respectively. For overall description of the Digital Library components, refer back to Chapter 1, "IBM Digital Library Overview" on page 1.

5.2.1 Hardware and Software Required

Table 5 lists the hardware requirements for installing and running the IBM Digital Library components on Windows 95. A CDROM drive is required for the program install only.

Table 5. Required Hardware				
Processor	**Memory**	**Storage**	**Display**	**Others**
Minimum: 80486DX **Recommended: Pentium Processor**	Minimum: 24 MB Recommended: 32 MB	Minimum: 160 MB Recommended: 512 MB	VGA	• Network Adapter • CDROM drive (for install only) • ASPI-compliant SCSI adapter for scanning

Table 6 lists the software requirements for installing the Digital Library Client for windows.

Table 6. Required Software			
Operating System	**TCP/IP Support**	**Development Environment**	**Digital Library Client**
Windows 95	Installed with Windows 95	• MS Visual C++ 4.0 • Borland C++ 4.51	VisualInfo Client for Windows (within IBM Digital Library 1.0 CDROM)

The VisualInfo Client for Windows supports the following scanners to import document images:

- Abaton 300s, GS, Color, Transcribe for Windows
- Agfa Scanners with Adaptec, Studioscan and Studioscan II
- Apple One Scanner, Apple MardiGras Scanner
- AVR (HP Emulation)
- Bell & Howell 500F, RSC(SCSI)
- Canon IX-12/30F/4015, Microfilm Reader, DR-100/3020, GP55F
- Chinon DS-3000
- Complete PC Scanner
- Datacopy 730 and 730GS with ASPI
- DocIt 3000/4000 Scanner
- Envision ENV 6100/8100/24 Pro
- Epson GT-4000/6500
- Fujitsu SP10, SP Jr., and M3096Gm
- M3097Gm (Windows NT)
- HP SCANJet (Windows NT)
- Howtek Personal Color Scanner

- IBM PageScanner, Adapter 3119/A
- IBM 2456 (Windows NT)
- JetFax4
- KOFAX Kipp 2.0
- Microtek SM-IIhr, SM-III and ScanMajer II
- Nikon SCANTOUCH
- Panasonic FX-RS 307/505/506 and SCSI
- Pentax IQ Scan (HP Emulation)
- Relisys series scanners
- Ricoh FS-2, FS-X IS-50/60/410/420/510/520
- Seaport scanners
- Sharp JX-300/320/325/330/450/610 and EF-3000s TDC 2610 through SCSI
- UMAX Scanners
- Xerox X3002
- Xionics supported scanners (Simplex only)
- ZSOFT DEST PCScan Plus 1000/2000/3000

The client application for Windows supports all APA printers supported by Windows 95.

5.2.2 Before Installation

This section gives some background information about the installation of the VisualInfo Client for Windows including prerequisites. Before installation, check the following:

- It is assumed that you have Windows 95 available with your PC and you are familiar with some basics about interacting with Windows 95, such as running a program and dealing with a dialog box.

- Client for Windows supports only TCP/IP communications. It does not support SNA communications.

- Although only the Library Server is required to have been installed before the Client for Windows installation, we recommend that DL Admin and Object Server also be installed first, for verifying some client operations after the installation of the client program.

- The type of operating system, server name, port number and host name of your Library Server will be needed for the configuration of the client connection with Library Server during installation. Consult with DL Admin before you install Client for Windows.

There are four types of selectable components you can choose to install:

- **Library Interface** is a set of APIs that allow the user to connect to the Library Server and Object Servers.

- **Client Application (VisualInfo)** is an end-user interface to the Library Server for document or folder management, porting and exporting documents, and searching for and processing documents.

- **Image Services** supports scanning, display, and print.

- **Development Environment (Toolkit and Samples)** lets you create applications using the Digital Library APIs and user exits. It also contains the header and library files you need to compile and link-edit your application programs and sample code.

Based on your system needs, you can install these options selectively. Default is that you install all the components. You will be prompted at Step 5 below.

5.2.3 Installing the Client for Windows

The installation procedure is as follows:

1. In the initial desktop window of Windows 95, load IBM Digital Library CDROM in the CDROM drive (*D* or *E*) and click **My Computer**.

2. Click the CDROM icon entitled **DLenuv1** in the **My Computer** window (Figure 32).
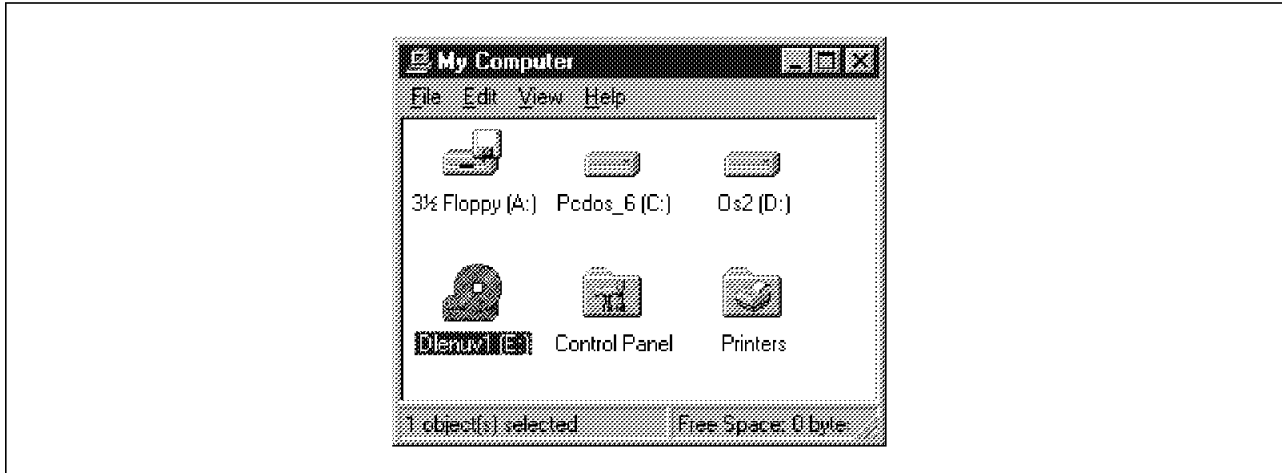
Figure 32. My Computer Window.

3. Click the **DLwin** icon, which produces the window shown in Figure 33.
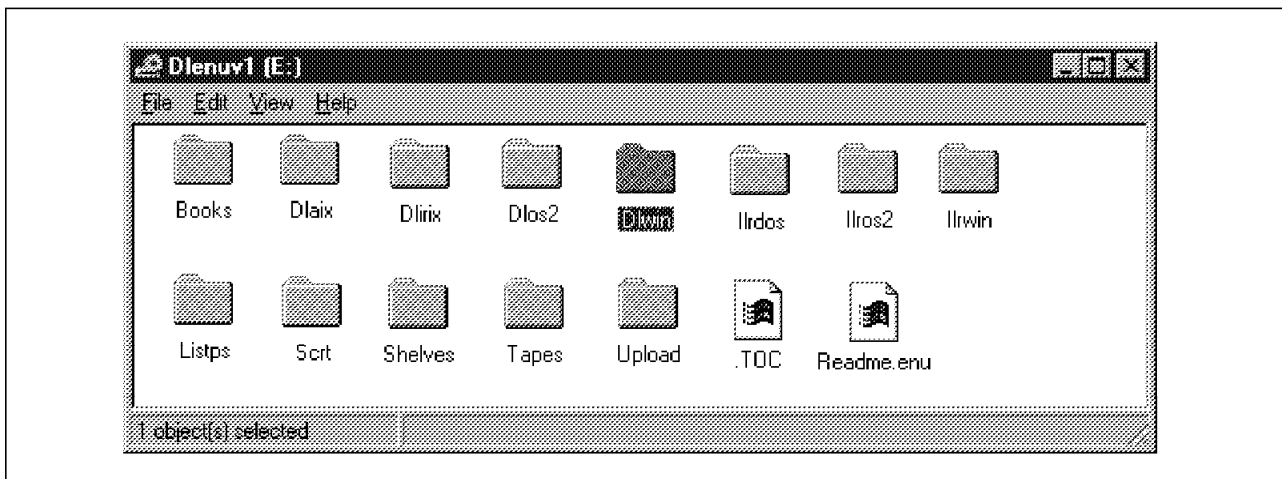
Figure 33. DLenuv1 window. This window shows the IBM Digital Library programs.

4. In the DLwin window (Figure 34 on page 55), click the **Setup** icon to start the program.
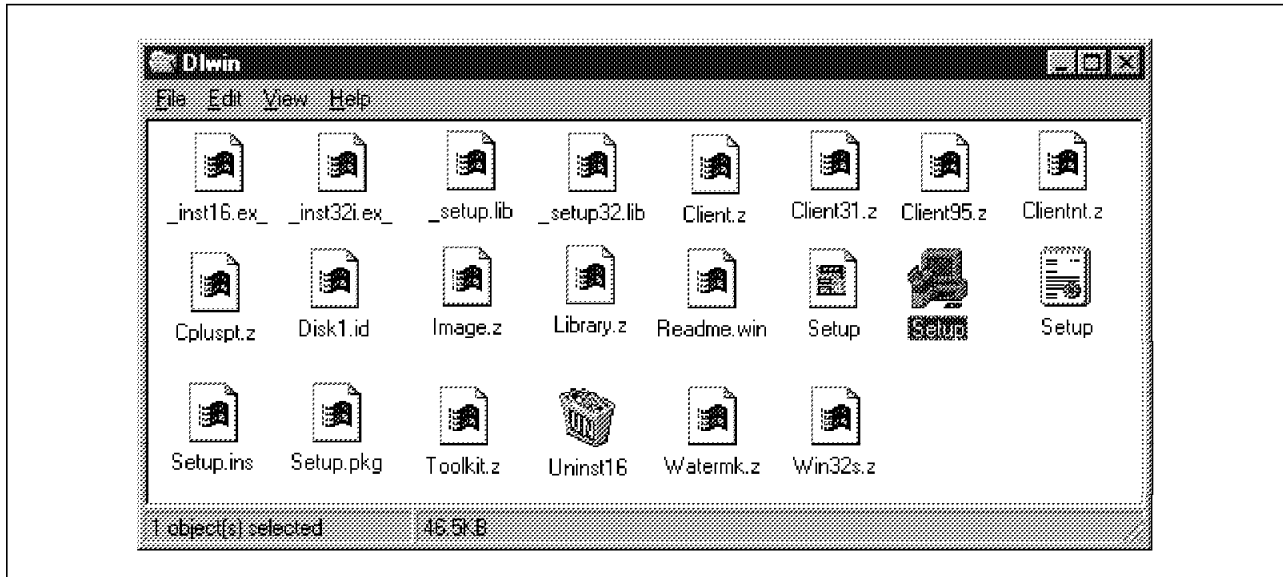
Figure 34. Dlwin window. This window shows the Client programs for Windows.

5. Click the **Next** buttons in the following windows until you are given the **Type of Installation** window.

   There are three choices: Standard Client, Selectable Components and Development Environment (see Table 7 for description)

Table 7. Install Choices for Windows				
Install Choice	**Library Interface**	**Client Application**	**Development Environment**	**Image Services**
Standard Client	Default	Default	-	Default
Selectable Component	Selectable	Selectable	Selectable	Selectable
Development Environment	Default	Default	Default	Default

The description of the components is given in the beginning of this section. We recommend you choose either Selectable Components or Development Environment for a loophole-proof installation. (We choose Selectable Components.)

6. Click the **Selectable Components** button.
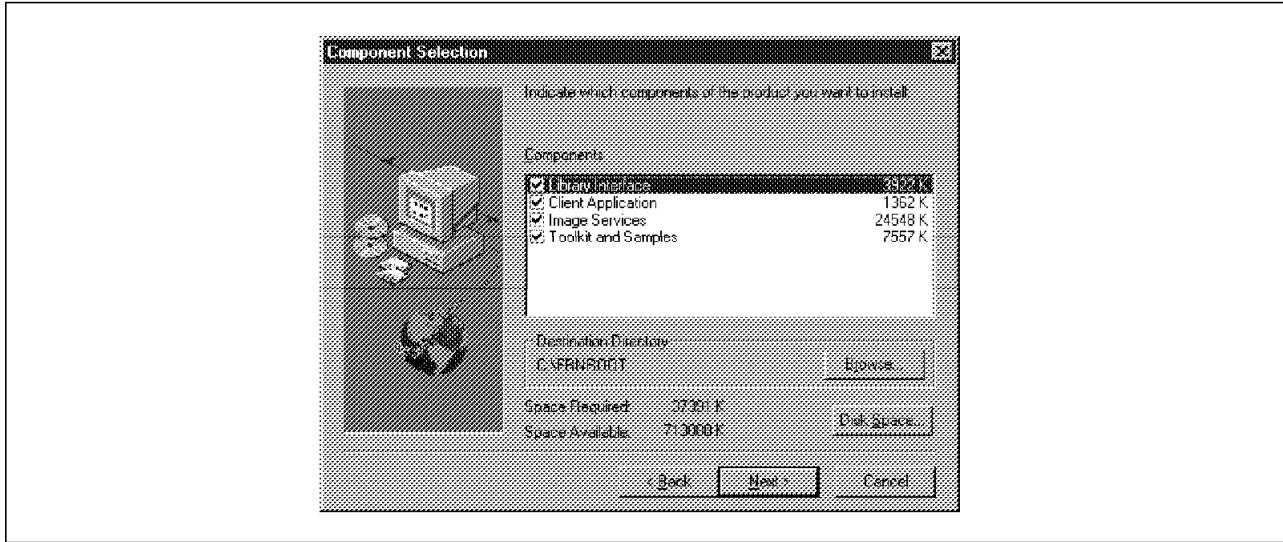
Figure 35. Component Selection window

In the Component Selection window (Figure 35), select the components you want to install. (We choose the default selection, with all the components checked.) Select the destination directory to install the selected components. (We assume the default drive *C*. Other drives and directories can be chosen using the **Browse** button.)
Click **Next** to continue.

7. In the Folder Selection window (Figure 36), select the folder name. (We assume the default folder name. Type a folder name if you want to change the name of the program folder.)

Click **Next** to continue.
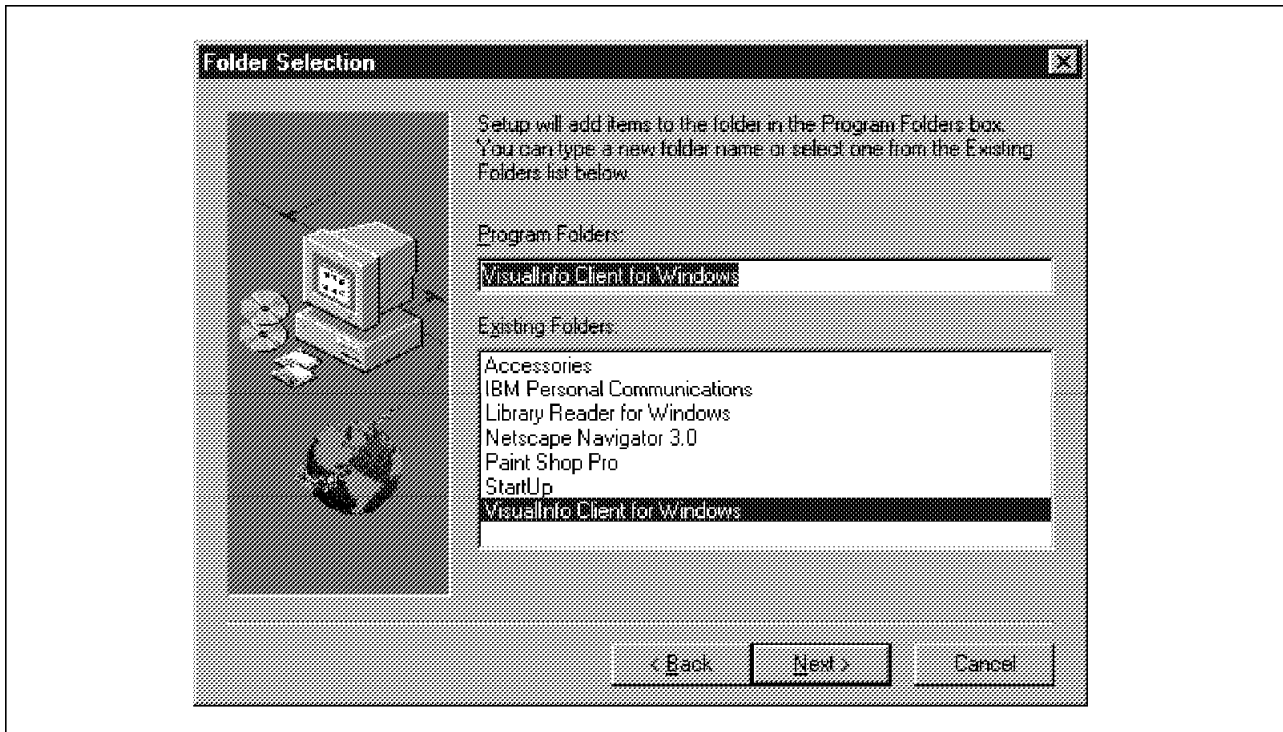
Figure 36. Folder Selection window.

8. In the **Start Copying Files** window (Figure 37 on page 57), review the settings and choices for the installation.
Click **Next** if you are satisfied with the specified settings and choices.
Otherwise, click **Back** to go back and change your choices.
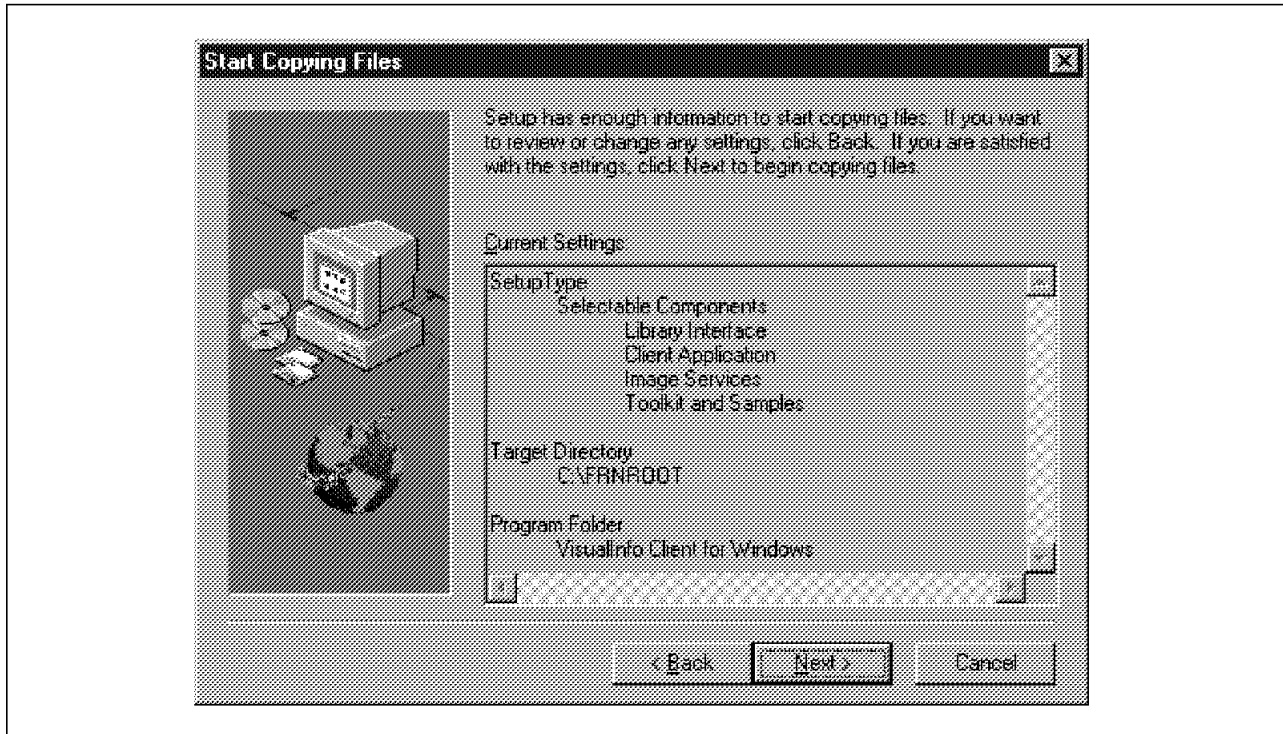
Figure 37. Start Copying Files Window

9. The **Setup** window displays the progress of the program installation.

10. The **Network Table Generation** window (Figure 38 on page 58) appears when the setup is 97% complete. You have two choices: to copy an existing network table or generate a new table. (We assume the latter.)
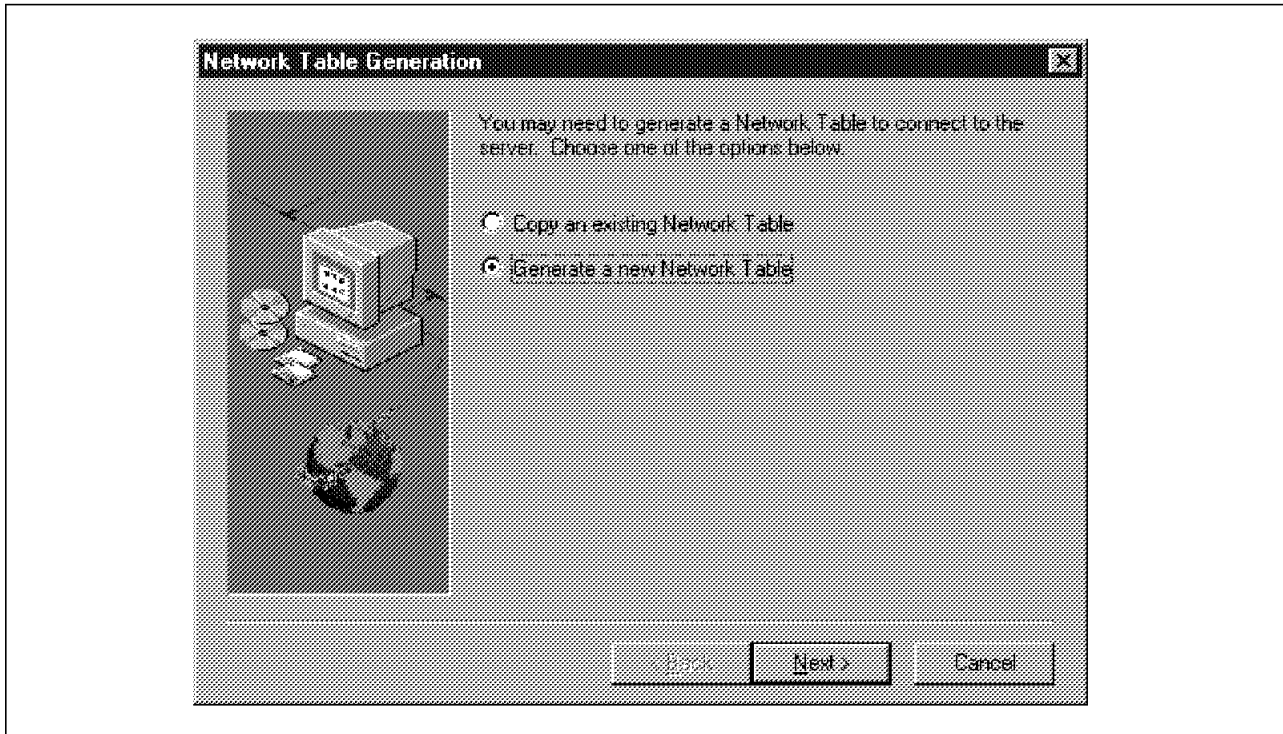
Figure 38. Network Table Generation Window

To generate a new network table, select **Generate a New Network Table** and click **Next**. Then, the Network Table Generation dialog opens (Figure 39 on page 59). Specify the following information about your Digital Library network: (Remember that the IBM Digital Library Client interacts only with the Library Server. See Chapter 1, "IBM Digital Library Overview" on page 1 for the description.)

- Operating system of the Library Server (we choose AIX)
- Name of the Library Server
- Port number
- Host name

Consult with DL Admin for the information above. If the network table file you create during installation does not work, you can modify it later using the "Network Table Generation Utility for Windows" (see 5.3, "How to Use the Utilities of VisualInfo Client for Windows" on page 60).
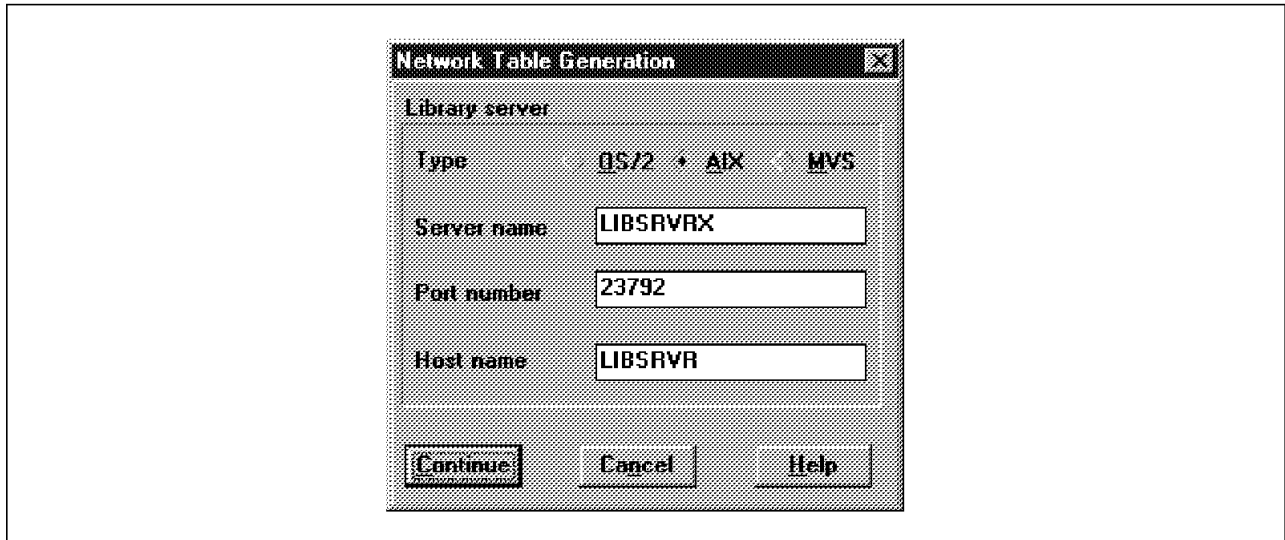
Figure 39. Network Table Generation Window for Creating a New Network Table

11. Click on **Continue** when you have finished specifying the information in the table.

    Click **OK** in the Information window to confirm the successful generation of a network table file.

12. The installation program will prompt you to modify the AUTOEXEC.BAT file.

    Click on **Yes** and **Ok** to confirm.

13. The Setup Complete dialog screen (Figure 40) allows you to restart your PC.

    Click on **Finish** button.
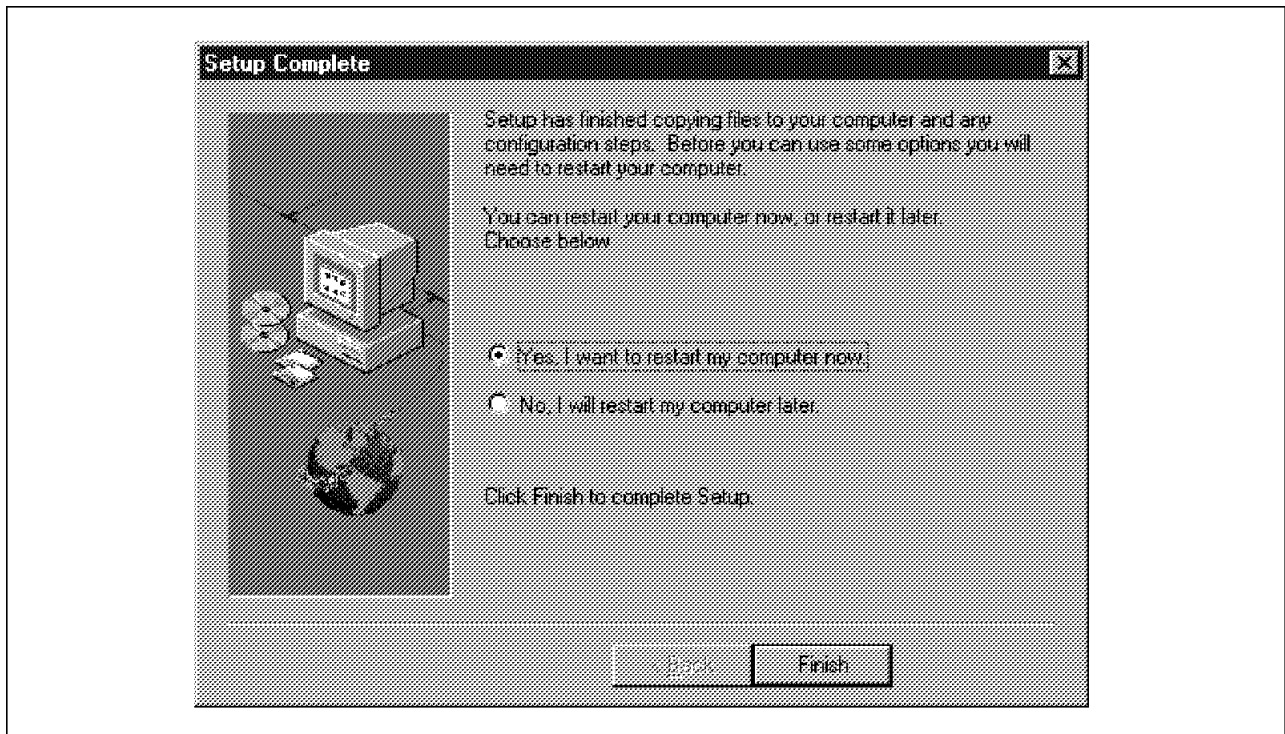
Figure 40. Setup Complete Window. The Digital Library installation is complete.

5.2.4 Making Digital Library Icon

If you want to add an icon for the installed program to facilitate an easy start of the client application, do the following:

1. From the desktop window of Windows 95, click the **Start** menubar, select **Settings** and then **Taskbar**.

2. In the **Taskbar Properties** window, select **Start Menu Programs** and click on **Advanced**.

3. Click on **Programs** and double-click on **VisualInfo Client for Windows** program.

4. Highlight the executable file **Client Application**, drag it out of the window using the right-hand mouse button, and release. This opens a menu window.

5. Double-click on **Create Shortcut(s) Here**. An icon has been created with the name of "Client Application."

6. If you want to change the icon name, click the icon with the right-hand mouse button, select **Rename**, and type a new name.

5.3 How to Use the Utilities of VisualInfo Client for Windows

When you install the client application for Windows, you also automatically install several utilities that help you set up network table, remove client applications, and verify communication with the Library Server.

The **Network Table Generation** utility stores the network values needed for network communications in TCP/IP between Client for Windows and the Library Server. With this utility, you can change the network connection with the Library Server at any time so that the reinstallation of the whole client application can be avoided.

1. From the desktop window, click the **Start** menubar, select **Programs**, **VisualInfo Client for Windows**, and then **Network Table Generation**.

2. Check operating system, server name and port number of Library Server. Modify if needed.

The **Connection Verification** utility automatically verifies that the required communications sequence can be successfully established between the Client and the Library Server.

1. From the desktop window of Windows 95, click the **Start** menubar, select **Programs**, **VisualInfo Client for Windows**, and then **Server Connection Verification**.

2. Click the **Connect** button to verify the connection with Library Server.

The **Uninstall** utility is useful when you want to remove the VisualInfo Client for Windows from your system. The procedure is as follows:

1. From the desktop window, click the **Start** menubar, select **Programs**, **VisualInfo Client for Windows**, and then **Remove VisualInfo Client**.

2. In the Confirm File Deletion dialog, click **Yes** to confirm the removal of all the client program from your PC.

5.4 Verifying the Program Installation

This section presents one way to verify the operability of the installed IBM Digital Library 1.0. Since this book is not intended to be the user's operation guide, we demonstrate a few basic manipulations of documents through Client for Windows in order to verify the correctness of the installation of the components and their interactions.

Starting the Client Application for Windows,

1. In Windows 95, either double-click on the Digital Library icon (if there is one), or select **Start**, then **Programs**, then **VisualInfo Client for Windows**, and then **Client Application**.

2. The Client for Windows Logon dialog box opens with the VisualInfo Client window (the main window of Client for Windows) as the background window (Figure 41). Log on by typing the User name and password. (**Notice**: Your log-in user name and password must be assigned beforehand by DL Admin. Also, the Library Server and Object Server must be in service.)
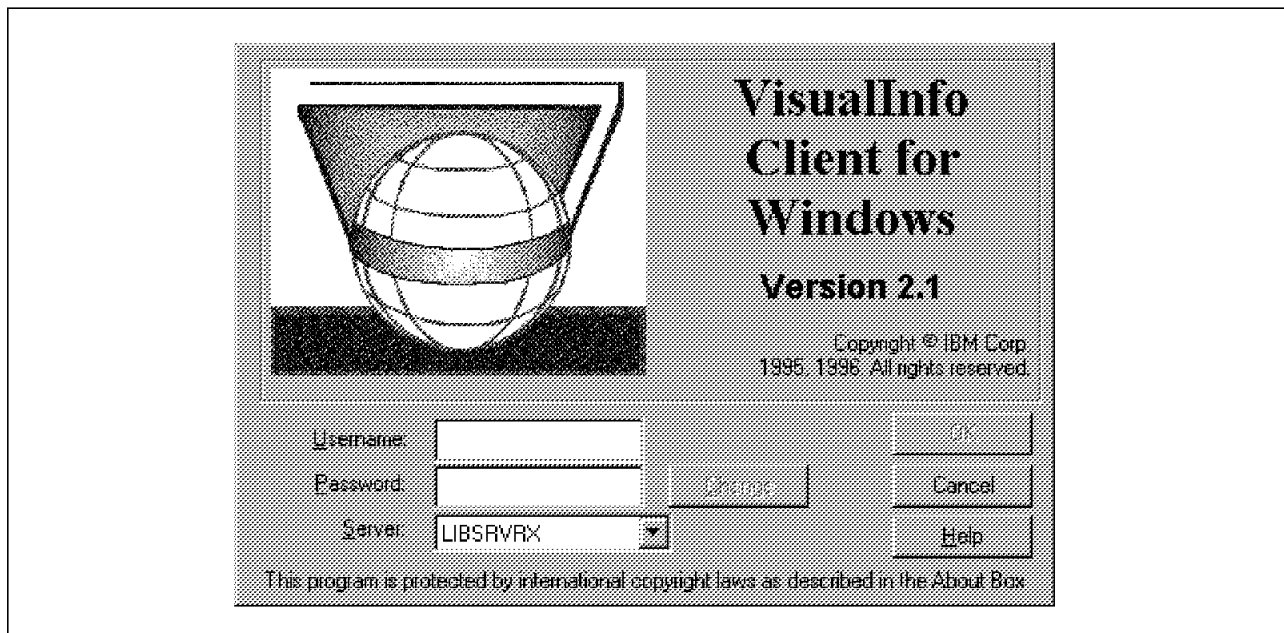
Figure 41. Logon Dialog Box

3. The **VisualInfo Client** window (Figure 42 on page 62) opens as the main window for all the client operations together with the **VisualInfo Viewer** window on the background. The main window is the starting place for all actions you perform with the Client for Windows, such as file handling (open, close, print, import and scan), document processing, and search.
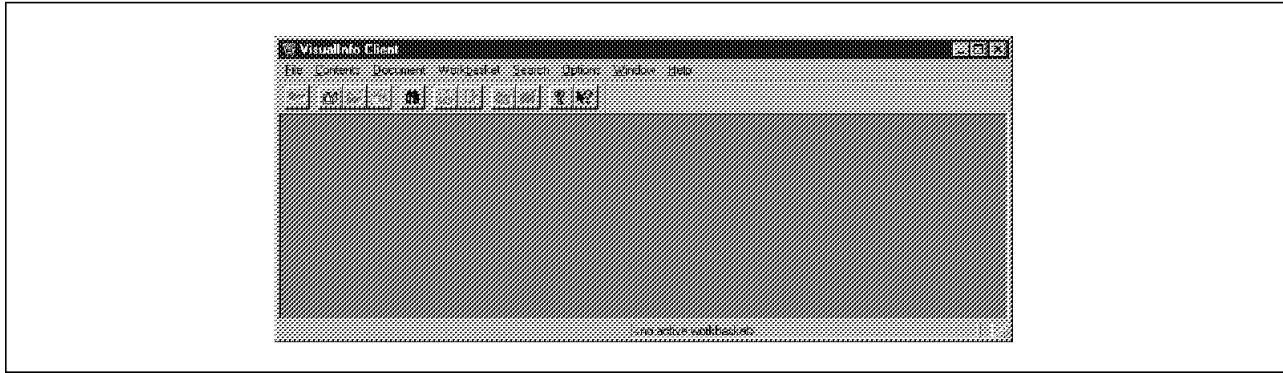
Figure 42. VisualInfo Client Window. The main window of Client for Windows.

5.4.1 Importing and Exporting Documents

You can import and export documents into the VisualInfo system from Client for Windows. For the purpose of verifying the program install, you can import or export an example document as follows:

1. Using the NotePad(or WordPad) of Windows 95, make a text file, and save as *test.txt*. (You can also import files from different directories on your PC or LAN. The system supports a variety of data formats for files that you want to import, and the DL Admin can define more.)

2. Select **Import** from the File menu. This opens the **Import Object** window (Figure 43).
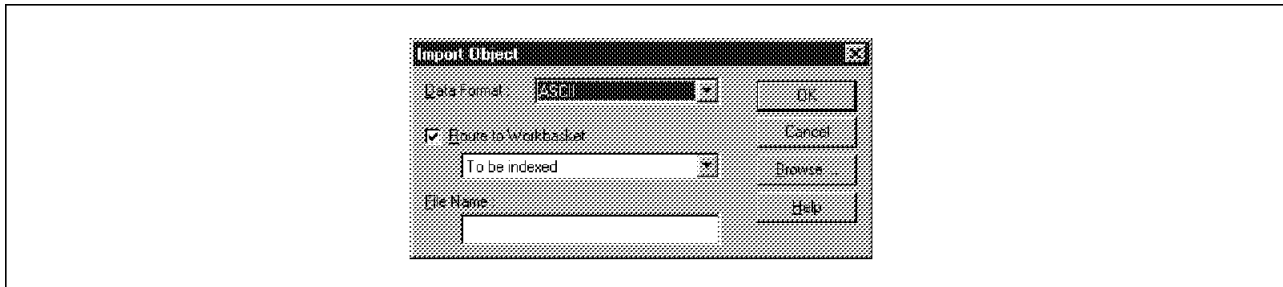
Figure 43. Import Object Window

3. Select ASCII as the data format of the object to be imported. Let's use the system-defined default workbasket called **To Be Indexed** to route the imported file to. Enter the complete path and the file name, *c:\Temp\test.txt* in this example. (Use the **Browse** button if you are unsure of the path and file name.)

4. To check the imported document, select **Open Workbasket** from the File menu of the **VisualInfo Client** window (Figure 42). Select **To Be Indexed** and click **OK**. Figure 44 on page 63 lists the documents in the workbasket.
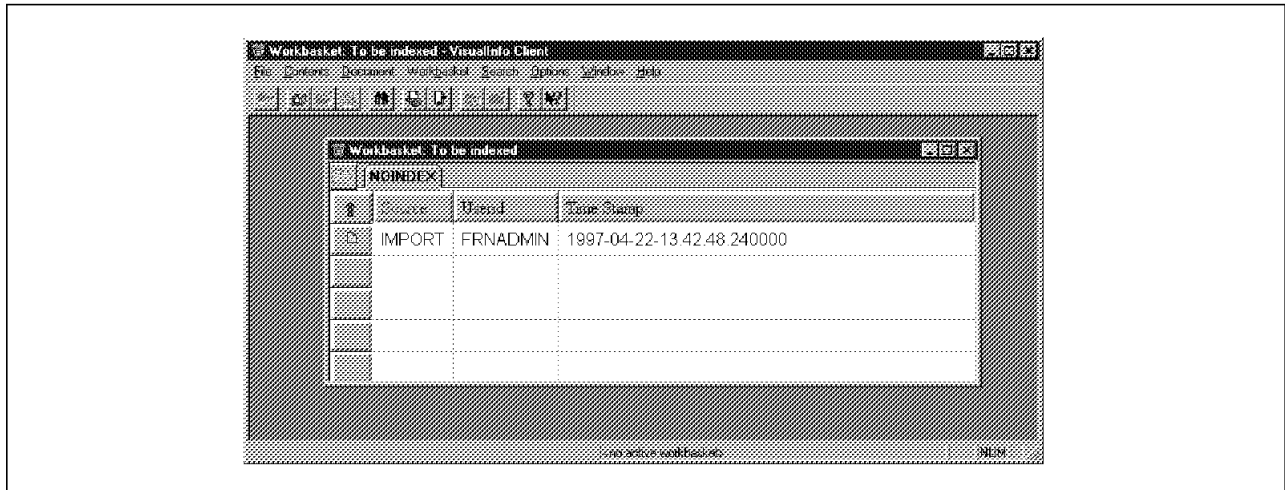
Figure 44. Workbasket Window.   Shows the contents of the workbasket "To Be Indexed"

     5. Double-click on the **IMPORT** file to view the content.

Also, you can export a document as follows:

     1. Highlight the document item in the Workbasket window (Figure 44) and then select **Export** from the Contents menu of the main window. The Export window opens (Figure 45).
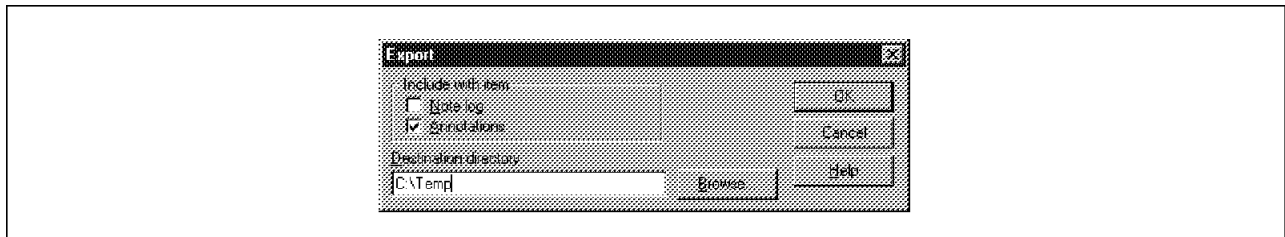
Figure 45. Export Window.

     2. Specify the directory path to export the document to, such as *C:\Temp*. (Use the **Browse** button if you are unsure of the path.)

     3. Click **Ok** to begin exporting the document.

     4. To check, use NotePad (or WordPad) of Windows 95 to view the file *Impo00xx* in *C:\Temp*.

5.4.2 Searching and Viewing Documents

We demonstrate how to search and retrieve a sample document that has been installed with Object Server using the specification of its index class and key field information:

     1. In the VisualInfo Client main window, select **Basic** from the Search menu. This opens the **Basic Search** window (Figure 46 on page 64).
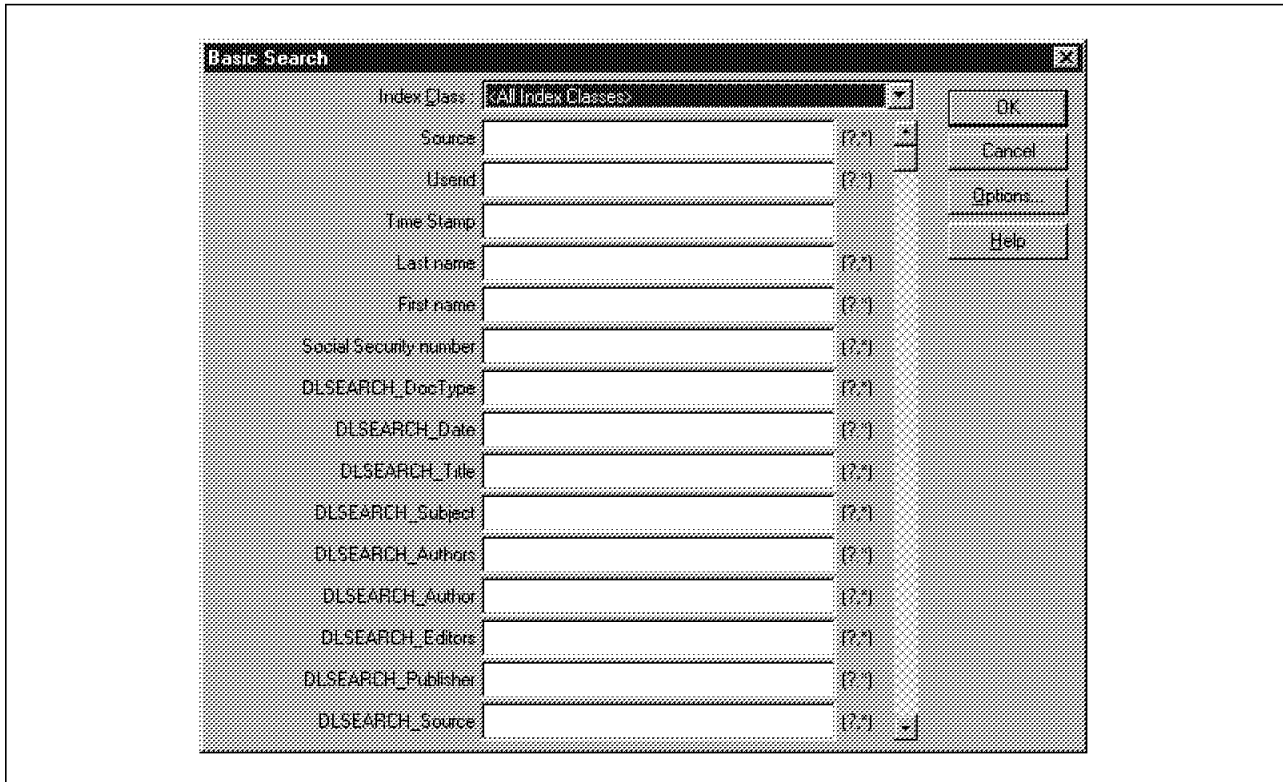
Figure 46. Basic Search Window. This window prompts Index Class and Key Field information

2. Select an index class **DLSEARCH_JOURNALS** from the drop-down Index Class list. This updates the **Basic Search** window as shown in Figure 47 on page 65.
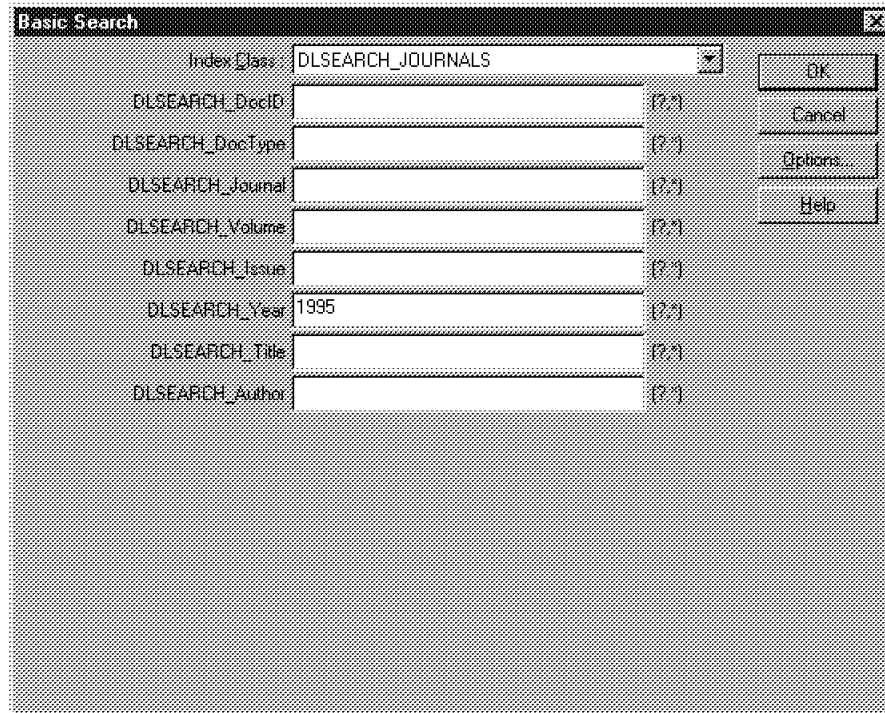
Figure 47. Updated Basic Search Window. This window prompts for the Key Field information of index class DL SEARCH_JOURNALS for search.

> 3. Type in 1995 and click **OK** button to show the search result. The Search Result window (Figure 48) lists the search result of DLRESEARCH_JOURNALS in 1995.
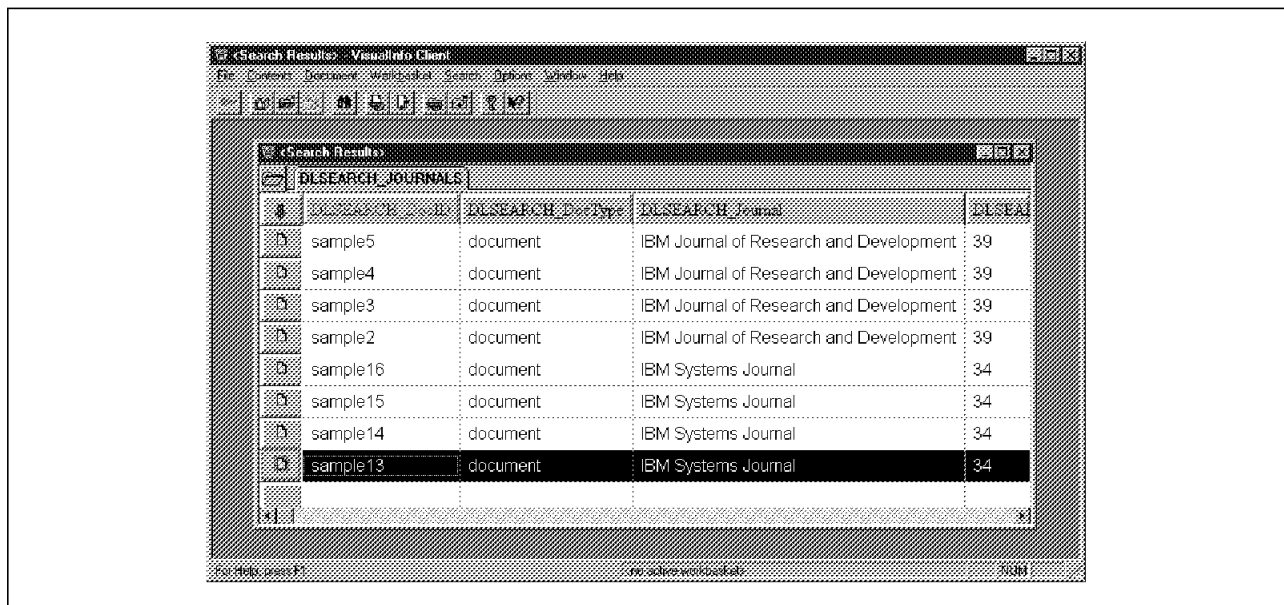
Figure 48. Search Result Window, Showing the Search Result of DLRESEARCH_JOURNALS in 1995

> 4. Double-click **Sample 13** to view the contents of the item.

For printing the document, select **Print** from the File menu on the **VisualInfo Viewer** window. Use **Printer Setup** selection to select a printer, set paper size and orientation.

For processing the object, use other menubars such as **Edit**, **View**, and **Window**.

Chapter 6. Customized Design of IBM Digital Library

This chapter examines a Digital Library solution with IBM Digital Library 1.0 and custom-modeled data. The main purpose of this other solution is to give you a greater understanding of the types of changes required when designing your own Digital Library solution. For demonstration purposes, we have chosen videos of IBM TV commercials as the custom-modeled data set. Since each customization will most likely imply a new data model, and possibly new requirements for accessing the data, we recommend referring to the Digital Library API Guides when implementing your own application.

Our customized system is based on the client/server configuration in which a Library Server with Cliette, an Object Server, and a DL Admin are connected by a LAN. In this solution, since customization is needed only for the Internet connection, we use Web users as the IBM Digital Library clients. For the description of the DL Admin, the Library Server with Cliette, and the Object Server, you can refer back to Chapter 2, "Digital Library Administration and Utilities on OS/2" on page 7, to Chapter 3, "Library Server and Cliette" on page 29, and to Chapter 4, "Object Server for AIX" on page 41, respectively. Figure 49 illustrates the system configuration.
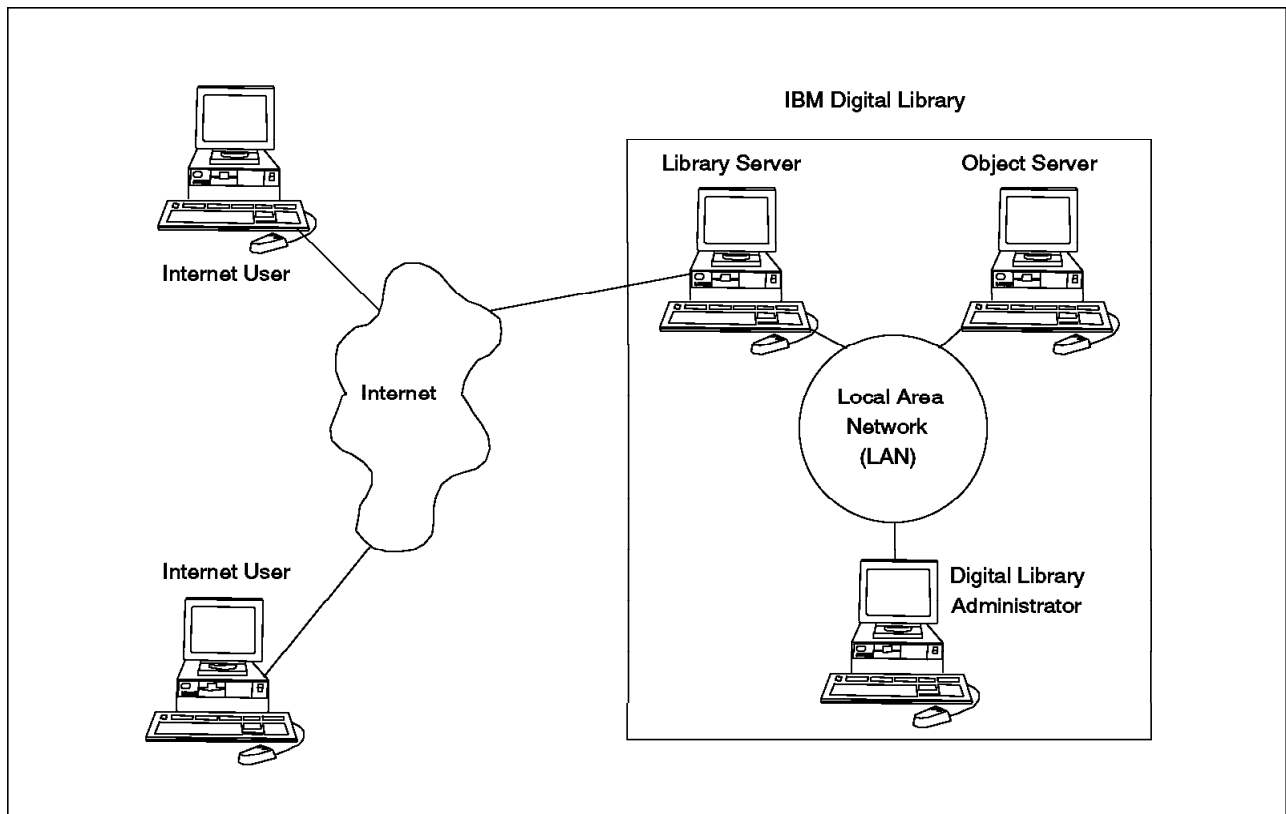
Figure 49. Configuration of the Customized Digital Library System

Table 8 on page 68 shows the general environment of our application.

Table 8. Customization Features				
Digital Library Components	**User Interface**	**Data Types**	**Search Indexes**	**Related Software Components to Customize**
Library Server with Cliette Object Server DL Admin Internet Client	HTML Web	MPEG Video Text GIF (with IBM TV commercials)	YEAR LANGUAGE DIALECT PRODUCT Query Terms for Text Search (all optional)	• Data Loader (in C) • Page Builder (in HTML and PERL)

Installing the customized IBM Digital Library solution requires modifying the following tasks in addition to the basic installation of the sample system shown in the previous chapters:

- Data modeling and loading

- Web page building for internet connection

Prerequisites: Proper understanding of the C programming for modifying the Data Loader Program (refer to 6.1, "Data Modeling and Loading" on page 69), and HTML and PERL programming for modifying Web pages and dynamic page builder programs (refer to 6.2, "Customizing the Internet Connection" on page 72).

Figure 50 illustrates a simplified workflow.
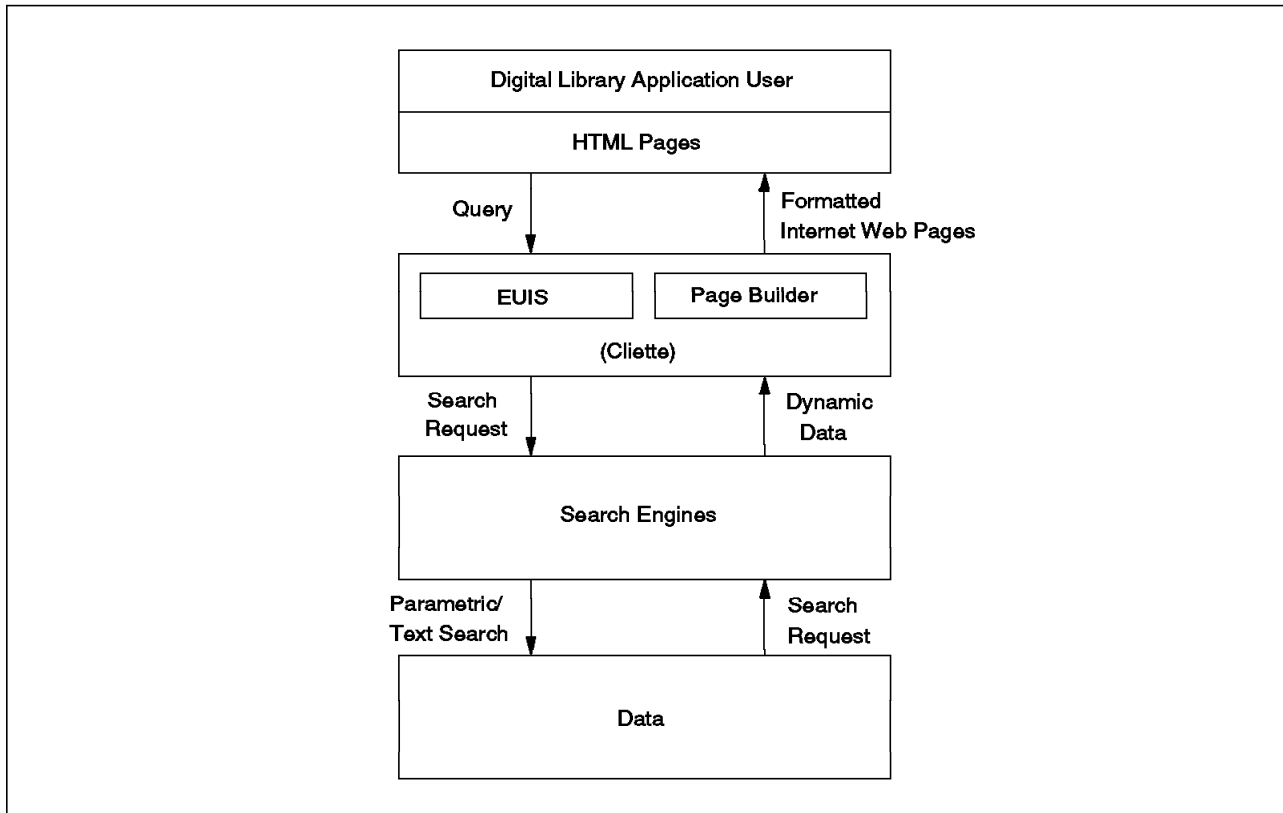
Figure 50. Simplified Data Flow via Internet Connection

We give the needed customization information in the following two sections.

The steps are these:

1. With the new index classes and attributes defined in Library Server and DL Admin, the Data Loader program loads the new data into Object Server. The data are stored, searched, and displayed according to a data model, which is a logical view of the organization of the data. The corresponding customization guide is given in 6.1, "Data Modeling and Loading."

2. Through the IBM Digital Library Web pages, a user's request with attributes and text (optional) is sent to Cliette, which then routes it to the search engines of the Library Server. The Object Server returns the search result to Cliette, which then dynamically forms the corresponding Web pages for displaying the result. The dynamic data are inserted into HTML form templates, which assembles the data into the Web pages. The corresponding customization guide is given in 6.2, "Customizing the Internet Connection" on page 72.

6.1 Data Modeling and Loading

For our data model, we use the data types of *MPEG* video, text and *GIF* (for thumbnail) with IBM TV commercials as customer-selected data. The steps to be followed in data loading are as follows:

1. Create key fields based on the attributes of the modeled data.
2. Create index classes using these key fields.
3. Create data control file in the Library Server.
4. Make the corresponding changes in the Loader program.
5. Load data using the Loader program.

6.1.1 Creating Key Fields

The IBM TV Commercial data consists of the advertisements of IBM products and services in different languages and dialects. Each commercial is given a title and a date, the year in which it was created. From this, we set up five major attributes: **Product, Language, Dialect, Title** and **Year**. In addition, two more attributes, **DocID** and **DocType**, have been added in order to distinguish different data when loaded. As a result, these seven attributes have been defined as key fields by DL Admin:

- TV_PRODUCT
- TV_LANGUAGE
- TV_DIALECT
- TV_TITLE
- TV_YEAR
- TV_DocID
- TV_DocType

The next step is to create index classes.

6.1.2 Creating Index Class

We have to create an index class called TV_COMMERCIAL with the above-listed key fields. The attributes **TV_TITLE** and **TV_DocType** should be set as the *required* key fields (see Figure 51 on page 70).
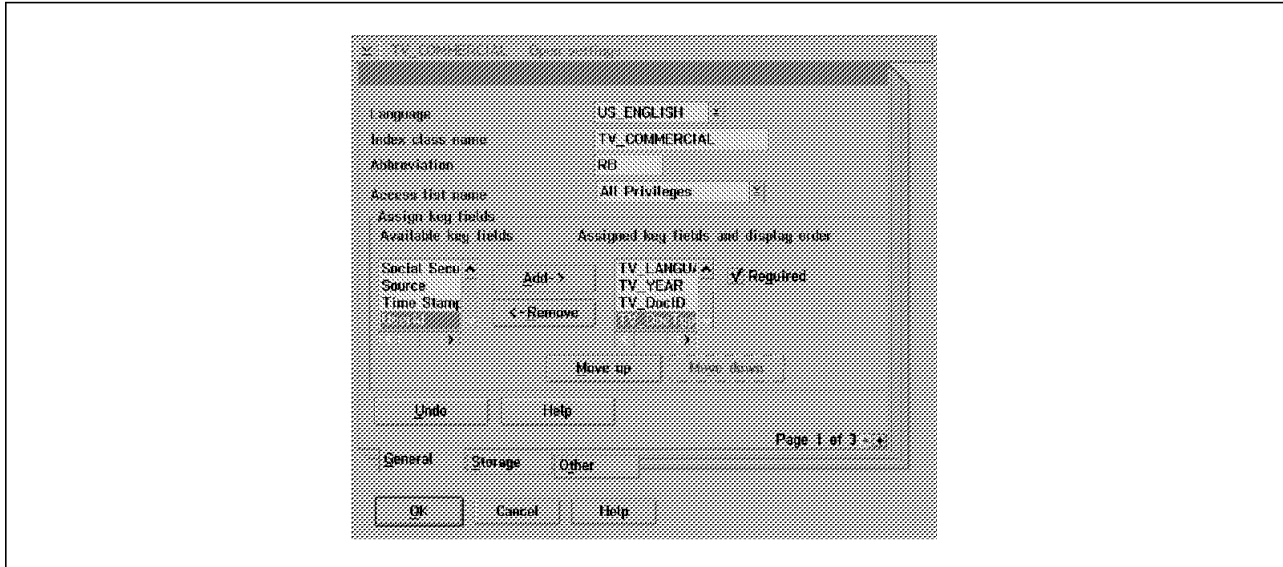
Figure 51. TV_COMMERCIAL Index Class Window

6.1.3 Creating a Control File

The Control File gives details about the data type to be loaded. The Loader program loads the data into the Digital Library according to the data types defined in the Control File. Follow the steps given below to create the Control File:

1. Log on to the Library Server as the *root* user and change directory to */usr/lpp/frn/samples/oqsdata*.
2. Create a directory for downloading the data (for example, *tv*).
3. Download all the data in this directory in the data types:
   - MPEG (Actual data, that is, TV commercials).
   - GIF (for thumbnail views).
   - Text (text about the commercials in ASCII format).
4. Create the Control File in this directory in the following format and add information for each commercial:

       Title
       Product
       Language
       Dialect
       Year
       GIF Image file
       MPEG File
       Text File

5. Save the Control File as *tv.dat*.

Next, we need to modify the Loader program to be able to load the custom-modeled data.

6.1.4 Loader Script and Program

Most of the steps remain the same as in the Sample System. The differences are pointed out here.

6.1.4.1 Loader Script

The loader script available in the sample system needs to be modified for our purpose. The loader scripts are *ldsample1* and *ldsample2* in the /usr/lpp/frn/samples/oqsdata directory. Go to this directory and do the following:

1. Copy *ldsample2* to *ldtv*.
2. Edit *ldtv* and change the line to the following.

   ```
   cd tv; ../frntv tv.dat tv_items LIBSRVRX > tv_err.txt ; cd ..
   ```

   where *frntv* is the loader executable, *tv.dat* is the control file, *tv_items* is the output file, **LIBSRVRX** is the Library Server database, and *tv_err.txt* is the error file. (Ensure that the path to all the files is correct and the respective directories have write permission.)
3. Save the Loader program.

We need to create an executable *frntv* the loader script can use to load the data in the system.

6.1.4.2 Loader Program

The Loader program is compiled from *C* source using a Makefile that is provided with Sample System. The first step is to copy *frnsaml2.c* to *frntv.c*. Then modify the Makefile to include a rule to make the *frntv* executable and a rule for compiling *frntv.c*. Follow the examples in the Makefile for creating these rules. The next and more important step is to change the C file to load the new data. We have accomplished this following the example in *frnsaml2.c*. The Loader program does the following:

1. Logs on to the Library Server.

2. Finds the ID of the class name.

3. Finds all the attribute IDs.

4. Ensures all the required IDs have been found.

5. Reads the data from the Control File.

6. Creates objects in the library, ensuring that each part is created separately according to its data type.

7. Repeats until all the data in the Control File have been read.

8. Logs off from the Library Server.

The changes made were basically to remove the attribute set for sample2 and add the new attribute set for our solution. The class name used is also of the new class for our solution. There are examples of how to load TEXT and IMAGES in the Sample System; the only new code that has been written is for loading MPEG data. Loading MPEG data uses basically the same as the existing call, except that the MPEG content class passed as a parameter should be set to 305.

Now, when a change is needed, it is only a matter of compiling and linking the Loader program using the Makefile. This can be done in the following way:

```
make frntv
```

6.1.5 Loading Data

To load the new data, follow the procedure below:

1. Log on as user *portal*.
2. Ensure that the Library Server and the Object Server are both running.
3. Change directory as follows:

        $ cd /usr/lpp/frn/samples/oqsdata

4. Make sure that the directory *tv* exists and all the data files are loaded.
5. Execute the following command:

        $ ./ldtv

   This loads all the data onto the Object Server.

Error is recorded in the tv_err.txt file.

For the index class to be accessible from the client, add the following line in the *frncliet.ini* (in the directory /usr/lpp/frn/samples/cliette) file under the **catalog** section,

        Catalog3=TV_COMMERCIAL TV_TITLE

This enables you to access a particular index class and its attributes from a Web client.

6.2 Customizing the Internet Connection

Digital Library's Internet Connection lets you build query applications for the World Wide Web. These applications accept input queries in a Web browser, search through Digital Library, and display results in the Web browser. Most search applications on the World Wide Web use one or two static query pages as the user interface, which thus cannot adjust to the user's input. IBM Digital Library Internet Connection extends the limits of ordinary HTML by dynamically updating the interface to match the user's actions. Internet Connection also formats the result pages dynamically using Page Builder.

Some Internet Connection components are supplied as object code and are accessed with APIs. Other components are supplied as source code and are accessed with subroutine calls; these source code components can be tailored to your needs. The custom-designed Internet Connection application we describe here is intended to show you how to tailor the Internet Connection applications of IBM Digital Library to your application's needs.

6.2.1 Modifying Programs

Once the data has been successfully loaded, you need to modify a few related programs for proper Internet connection and data representation. From the Library Server, identify the programs that are related to the customized system and modify them according to your needs. In the following, we describe the files used in our custom-configured library system:

- */usr/lpp/frn/samples/cliette/frncliette.ini* configures the Cliette by defining directory path to all the related files, host name, media type, search manager and data catalogs (for example, TV_COMMERCIALS).

- */usr/lpp/frn/samples/cliette/frnxwpb* is a PERL 4.0 source-code program called Page Builder which formats Web pages by making use of templates

(see below). The Page Builder works closely with a Cliette component called End-User Interface Services (EUIS).

- */usr/lpp/frn/samples/cliette/dlsearch.html* is the HTML home page file of our custom-designed Digital Library system. This Web page represents only the static data. Figure 52 shows the home page.
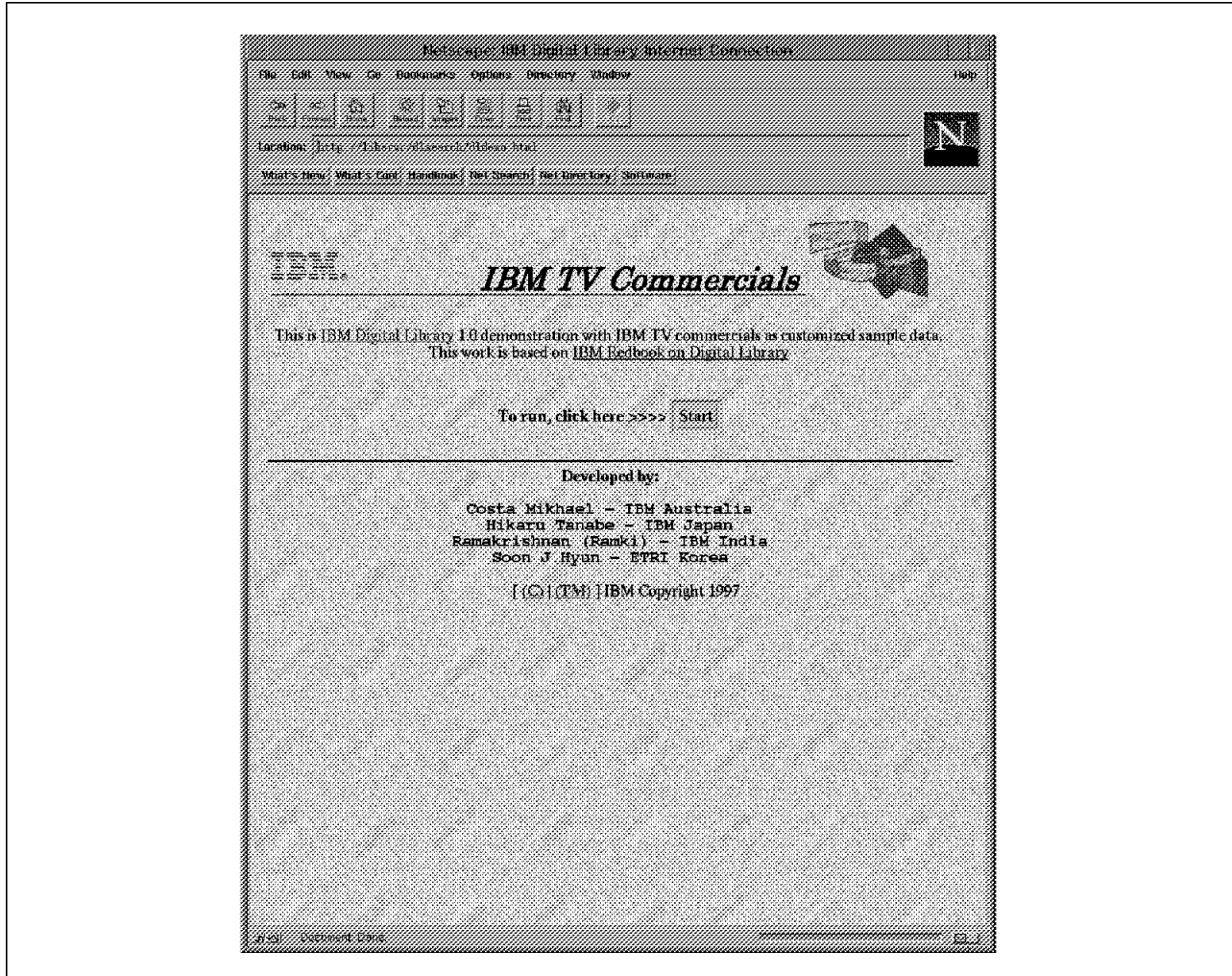
Figure 52. Home Page for TV Commercials

- */usr/lpp/frn/samples/cliette/*.wpb* are HTML files with imbedded PERL descriptions as templates. Templates are used by Page Builder in formatting the dynamic Web pages when the query results are retrieved from the search engines. The templates we modified in our experiment are described below with the resulting Web pages:

  - *attrib2.wpb* outlines forming a Web page called "Search Window" which enables the user to select attributes (such as YEAR, LANGUAGE, DIALECT, and PRODUCT) for parametric search, and to type in keywords (such as ThinkPad or music) for text search. Figure 53 on page 74 shows the corresponding Web page with *1997* as the user-selected attribute value to search all the IBM TV commercials made during the year.

Figure 53. Web Page for Library Search

     – *gallery.wpb* outlines forming a Web page called "Result Window," which displays the search result. Figure 54 on page 75 shows the search result.
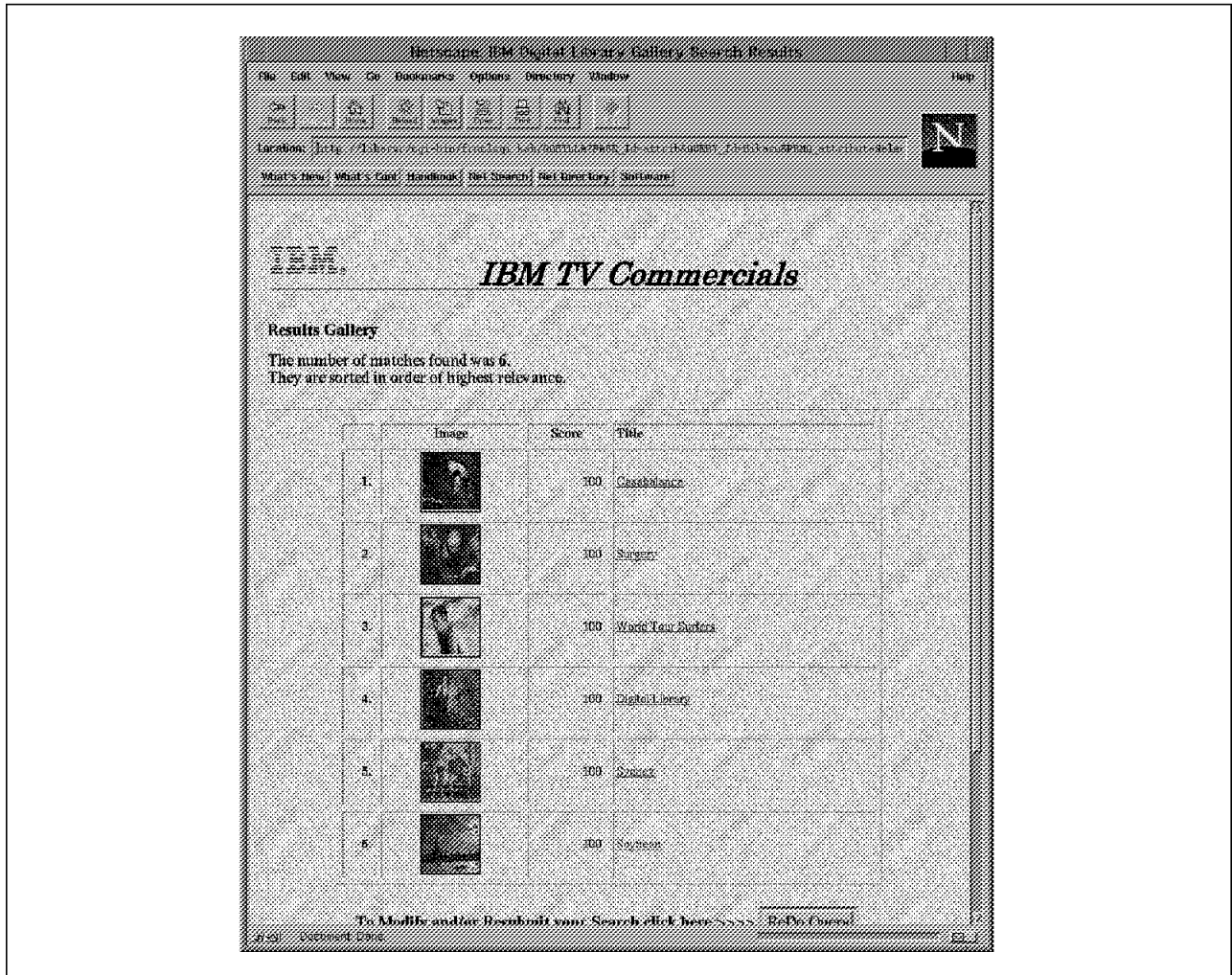
Figure 54. Web Page of Search Results

6.2.2 Viewing the Result

To view the video data you selected from the Result Window (Figure 54), you need an MPEG player program installed in your system:

- If you use Microsoft Internet Explorer as the Web browser, an MPEG video data can most probably be viewed automatically, since the Active Movie Player program is installed within the browser.

- For Netscape users, install an MPEG player. For example, the Active Movie Player program can be downloaded from the Microsoft home page.

Appendix A. Before Installation of Servers

A.1 Usage of SMIT

The **smit** command invokes the System Management Interface Tool (SMIT). SMIT is an interactive interface application designed to simplify system management tasks. The **smit** command displays a hierarchy of menus that can lead to interactive dialogues. SMIT builds and runs commands as directed by the user. Because SMIT runs commands, you need the authority to execute the commands that SMIT runs. To start SMIT, type following command from aixterm:

        smit
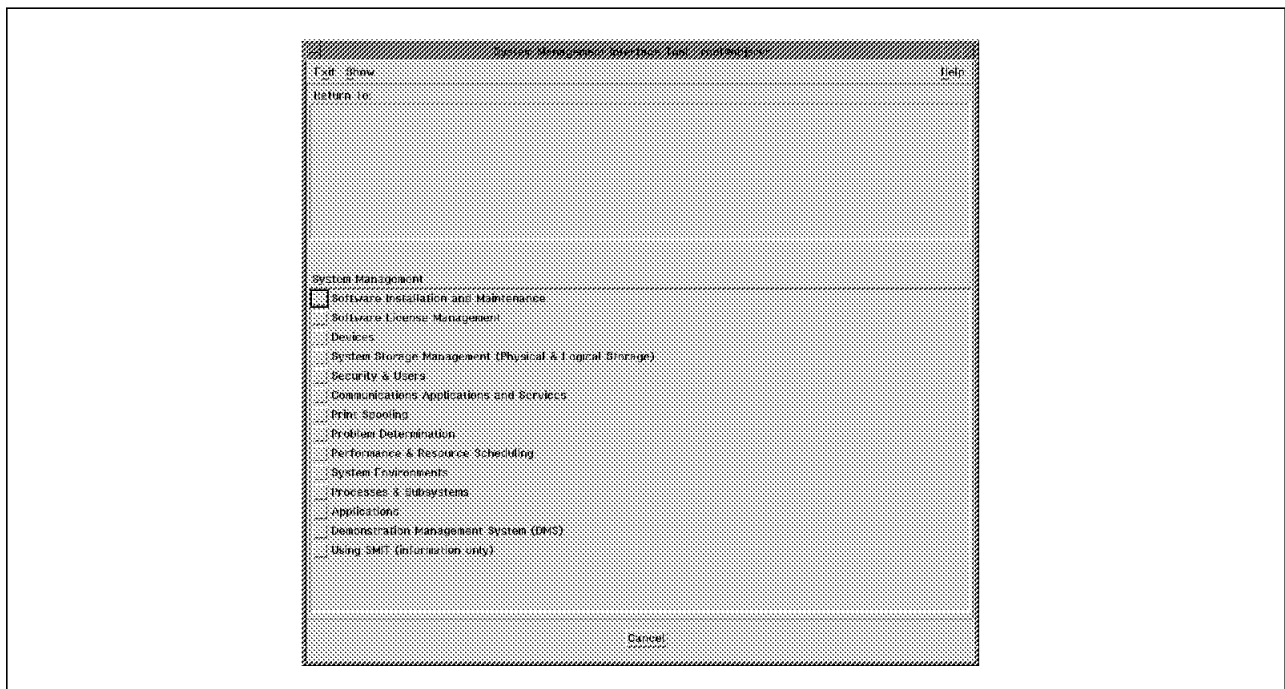
Figure 55. System Management Interface Tool Window

A.2 Setting Language Environment to IBM-850.

The code page installed in the default installation of AIX 4.1.5 is ISO8859-1. Digital Library requires a different code page, IBM-850 English (United States), on its AIX machines:

1. Log on as user root.

2. Type smit at the command prompt.

3. Open **System Environments** from the list of choices on the System Management panel (Figure 55).

4. Select **Manage Language Environment**.

5. Select **Change/Show Primary Language Environment**.

6. Select **Change/Show Cultural Convention, Language or Keyboard**.

7. Click on **List** and select **IBM-850 English (United States)** from the list for all the fields (Figure 56).

   Then click on **OK**.

8. Click on **List** of **INPUT device/directory for software** to select the device or directory from the list of available sources (usually */dev/cd0*) Then click on **OK**. Click **Done** when completed successfully.

9. Close SMIT.

10. Type the following command to shut down and reboot your system:

    # shutdown -Fr

```
Change/Show Cultural Convention, Language, or Keyboard : root@199

Primary CULTURAL convention          IBM-850    English (United States)    List

Primary LANGUAGE translation         IBM-850    English (United States)    List

Primary KEYBOARD                     IBM-850    English (United States)    List

INPUT device/directory for software  /dev/cd0                              List

EXTEND file systems if space needed? yes                                   List  ▲ ▼



    OK            Command            Reset            Cancel              ?
```

Figure 56. Change Code Page Window

A.3 Create/Mount File System

The following sections will help you in creating and mounting a file system in AIX.

A.3.1 Add Logical Volume

The steps are these:

1. Log on as user root.

2. Type smit at the command prompt.

3. Open **System Storage Management (Physical & Logical Storage)** from the list of choices on the System Management Panel.

4. Select **Logical Volume Manager**.

5. Select **Logical Volumes**.

6. Select **Add a Logical Volume** (Figure 57 on page 79).

7. Click on **List** and select a volume group name from the list.

8. Click on **OK**

9. Specify **Logical volume NAME**.

10. Specify **Number of LOGICAL PARTITIONS(Num.)**.

11. Click on **List** of **POSITION on physical volume** and change *middle* to *center*.

12. Click on **List** of **RANGE of physical volumes** and change *minimum* to *maximum*.

13. Click on **OK**.

14. Click on **DONE** when logical volume was created successfully.

15. Close SMIT.
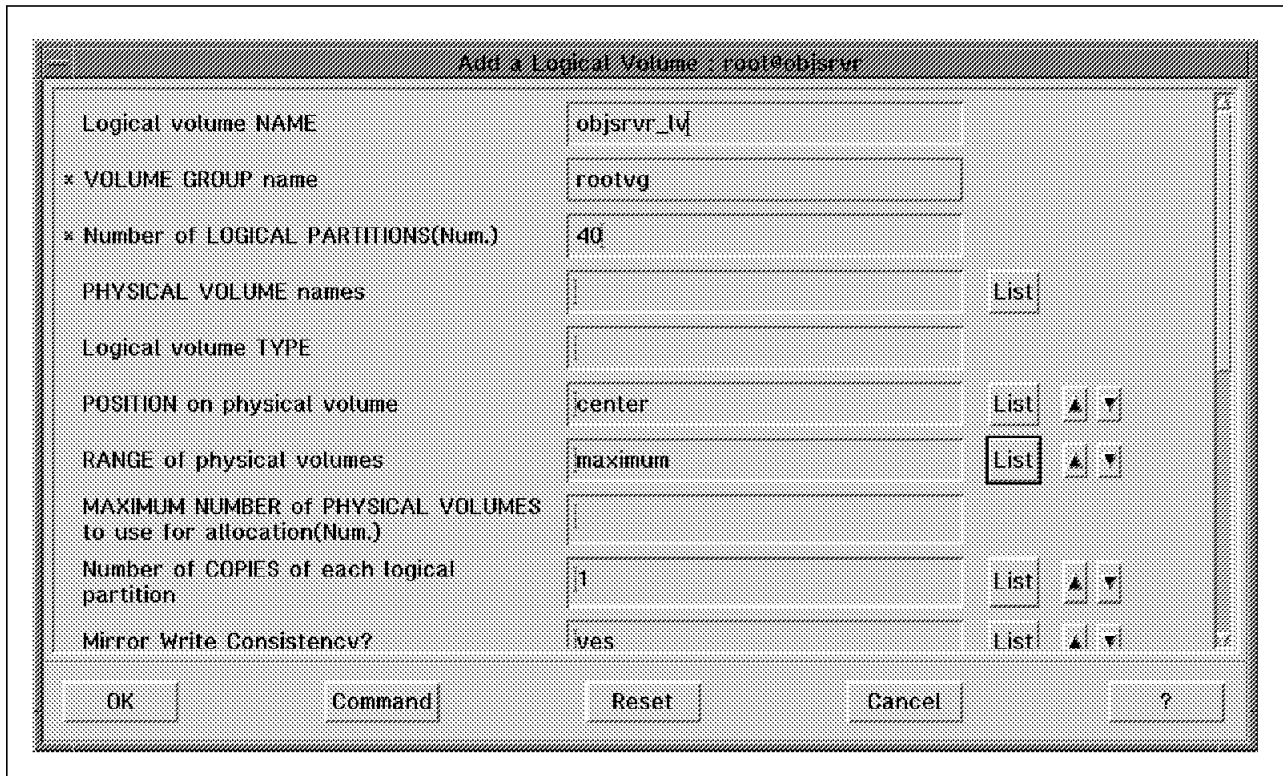
Figure 57. Add a Logical Volume Window

A.3.2 Add File System

Carry out these steps.

1. Log on as user root.

2. Type smit at the command prompt.

3. Open **System Storage Management (Physical & Logical Storage)** from the list of choices on the System Management Panel.

4. Select **File Systems**.

5. Select **Add / Change / Show / Delete File Systems** (Figure 58 on page 80).

6. Select **Journaled File Systems**.

7. Select **Add a Journaled File System on a Previously Defined Logical Volume**.

8. Click on **List** of **LOGICAL VOLUME name** and select logical volume name from the list.

9. Specify **MOUNT POINT**.

10. Click on **List** of **Mount AUTOMATICALLY at system restart?** and change *no* to *yes*.

11. Click on **OK**.

12. Click on **DONE** when file system is created successfully.

13. Close SMIT.
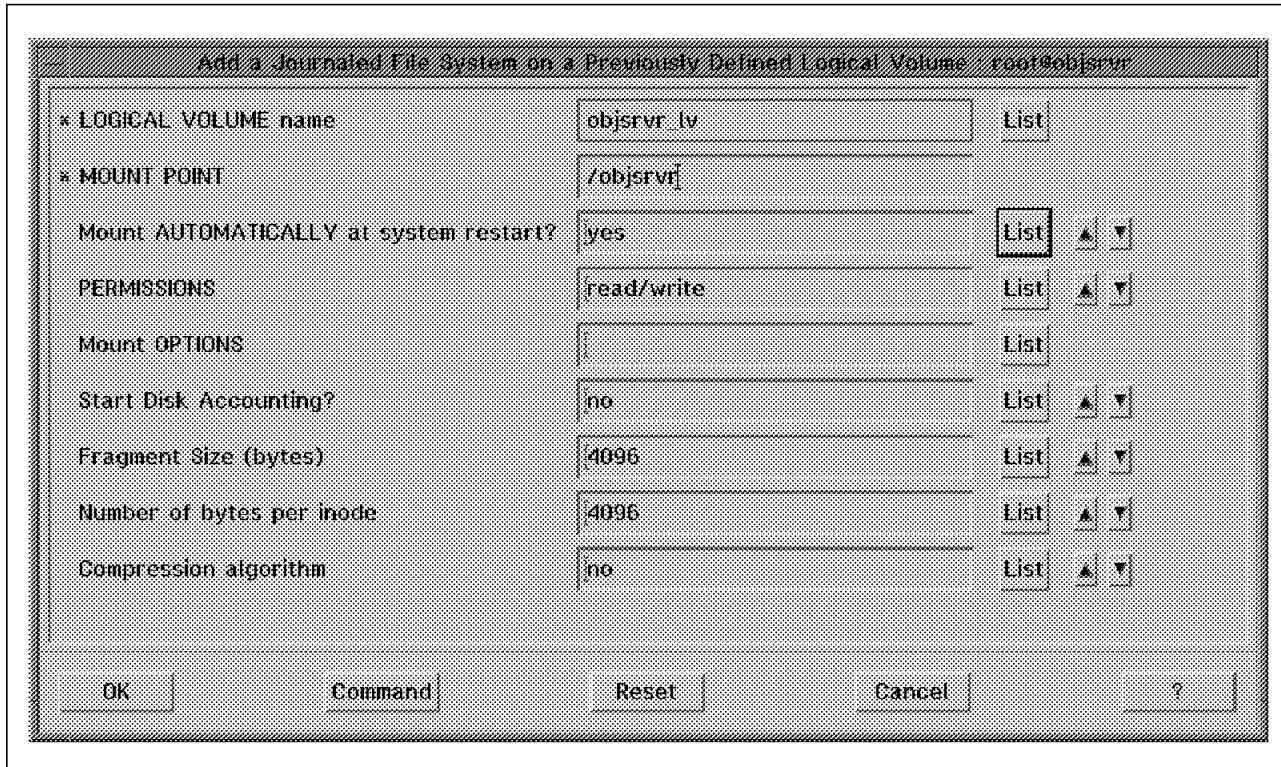
Figure 58. Add File System Window

A.3.3 Mount File System

The steps are these:

1. Log on as user root.

2. Type smit at the command prompt.

3. Open **System Storage Management (Physical & Logical Storage)** from the list of choices on the System Management Panel.

4. Select **File Systems**.

5. Select **Mount a File System** (Figure 59 on page 81).

6. Click on **List** of **FILE SYSTEM name** and select file system name from the list.

7. Click on **List** of **DIRECTORY over which to mount** and select mount point from the list.

8. Click on **List** of **TYPE of file system** and select *jfs* from the list.

9. Click on **OK**.

10. Click on **DONE** when file system is created successfully.

11. Close SMIT.
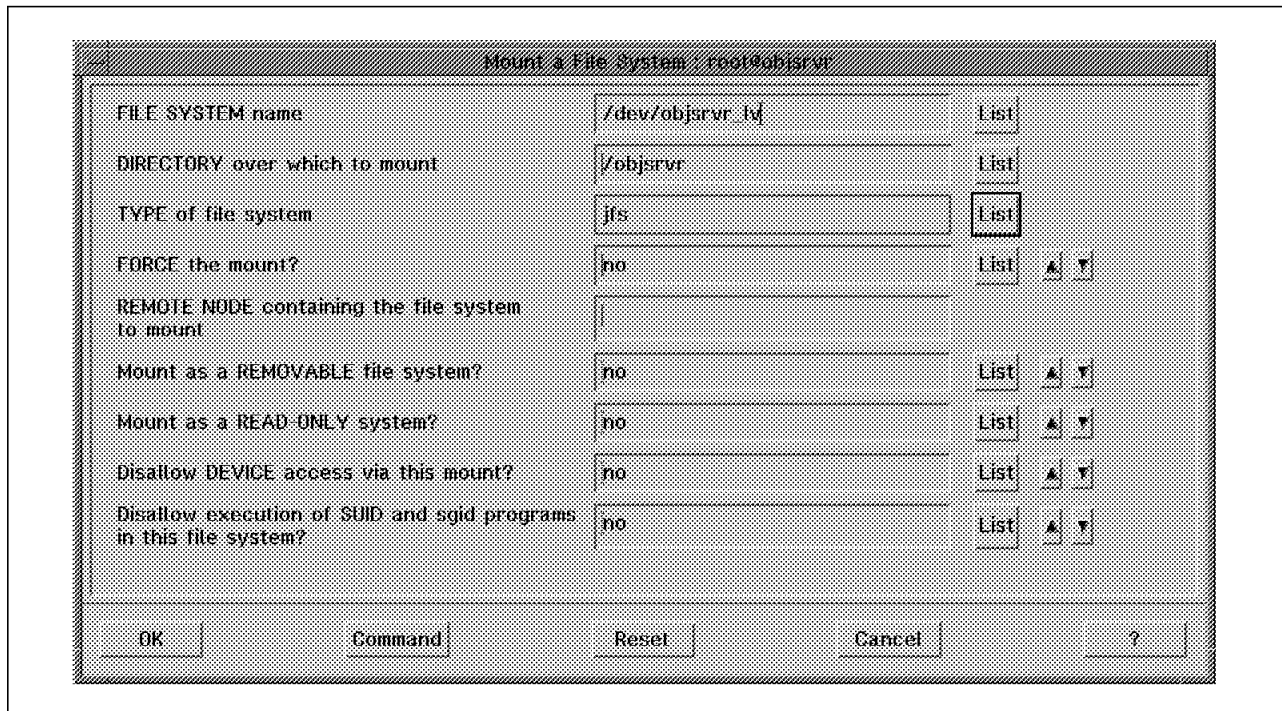
Figure 59. Mount File System Window

A.4 Create AIX User ID and Group

The primary group of the DB2 instance must be included in the group sets of the user ID that runs the Library Server and Object Server for AIX and the root user ID.

The library server and object server can be run by any defined user ID with the proper database authorization. However, the library server and object server run with root authority (as required for version AIX system calls). As a result, when the DB2/6000 database manager is used, all tables created by Digital Library are owned by user root.

This does not require the administrator who runs the library server and object server to have system root authority; the Digital Library administrator need only use a user ID that belongs to the DB2 instance owner's primary group. This is a requirement for proper operation of the library server and object server.

A.4.1 Add a Group

The steps are these:

1. Log on as user root.

2. Type smit at the command prompt.

3. Open **Security & Users** from the list of choices on the System Management Panel.

4. Select **Groups**.

5. Select **Add a Group** (Figure 60 on page 82).

6. Specify **Group NAME** (for example, *dba*).

7. Click on **List** of **ADMINISTRATIVE groups** and change *false* to *true*.

8. Click on **OK**.

9. Click on **DONE** when the new group is created successfully.

10. Close SMIT.
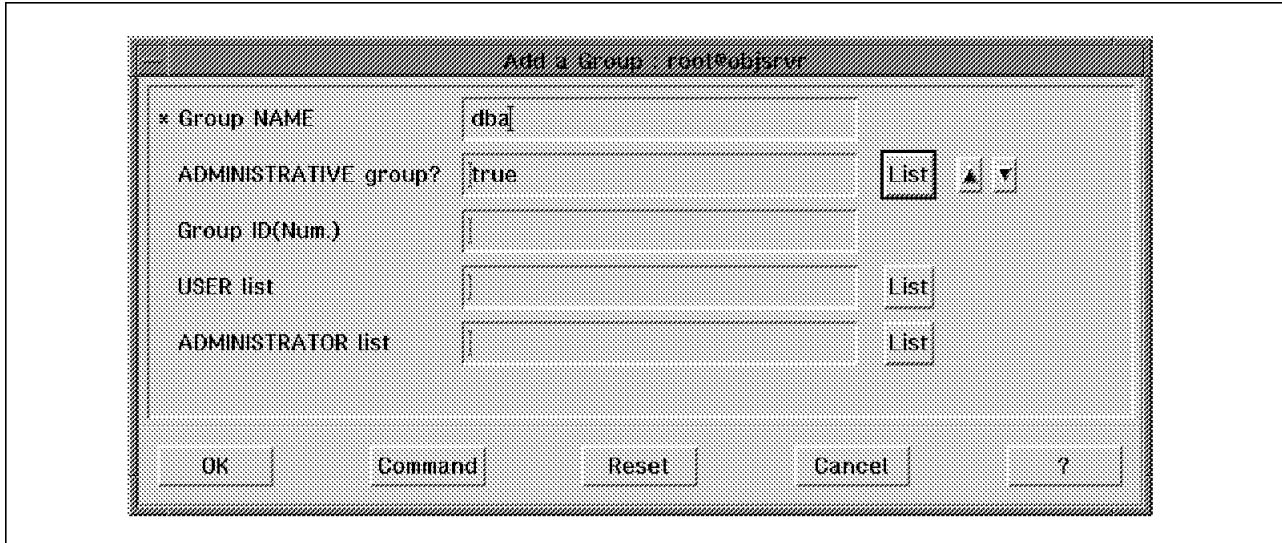
Figure 60. Add a Group Window

A.4.2 Change Settings of Root

The steps are these:

1. Log on as user root.

2. Type smit at the command prompt.

3. Open **Security & Users** from the list of choices on the System Management Panel.

4. Select **Users**.

5. Select **Change/Show characteristics of User**.

6. Click on **List** of **User NAME** and select **root** from the list (Figure 61 on page 83).

   Then click on **OK**.

7. Click on **List** of **Group SET** (see Table 9) and add created group name (for example, *dba, ...*).

8. Click on **OK**.

9. Click on **DONE** when the changing of setting of root is completed successfully.

10. Close SMIT.

Table 9. Group Set of Root	
UID	**Group SET**
root	dba, ...
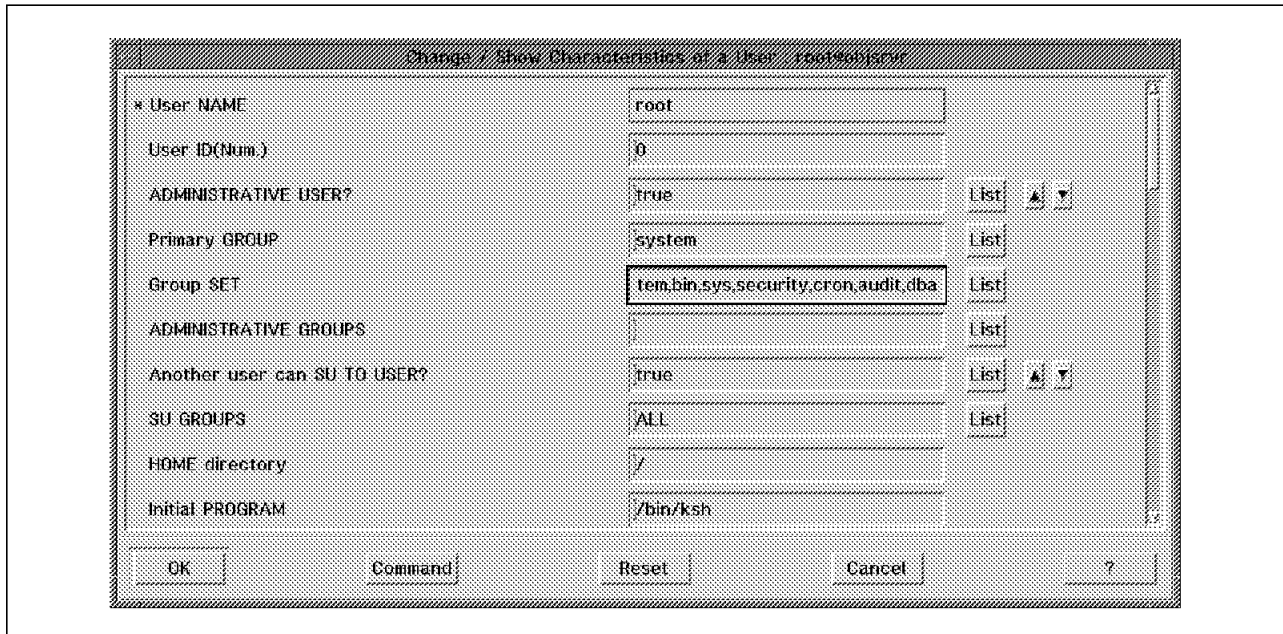
Figure 61. Change Settings of Root Window

A.4.3 Create User IDs

The steps are these:

1. Log on as user root.

2. Type smit at the command prompt.

3. Open **Security & Users** from the list of choices on the System Management Panel.

4. Select **Users**.

5. Select **Add a User** (Figure 62 on page 84).

6. Specify **User NAME**(for example, *objusr*).

7. Click on **List** of **ADMINISTRATIVE USER?** and change *true* to *false*.

8. Click on **List** of **Primary GROUP** and select *DB2 instance owner's primary group* (for example, *dba*) as shown in Table 10 on page 84.

9. Click on **List** of **Group SET** and select *DB2 instance owner's primary group* (for example, *dba*).

   Then click on **OK**.

10. Specify the **HOME directory**.

11. Click on **OK**.

12. Click on **DONE** when the object server user ID was is successfully.

13. Close SMIT.
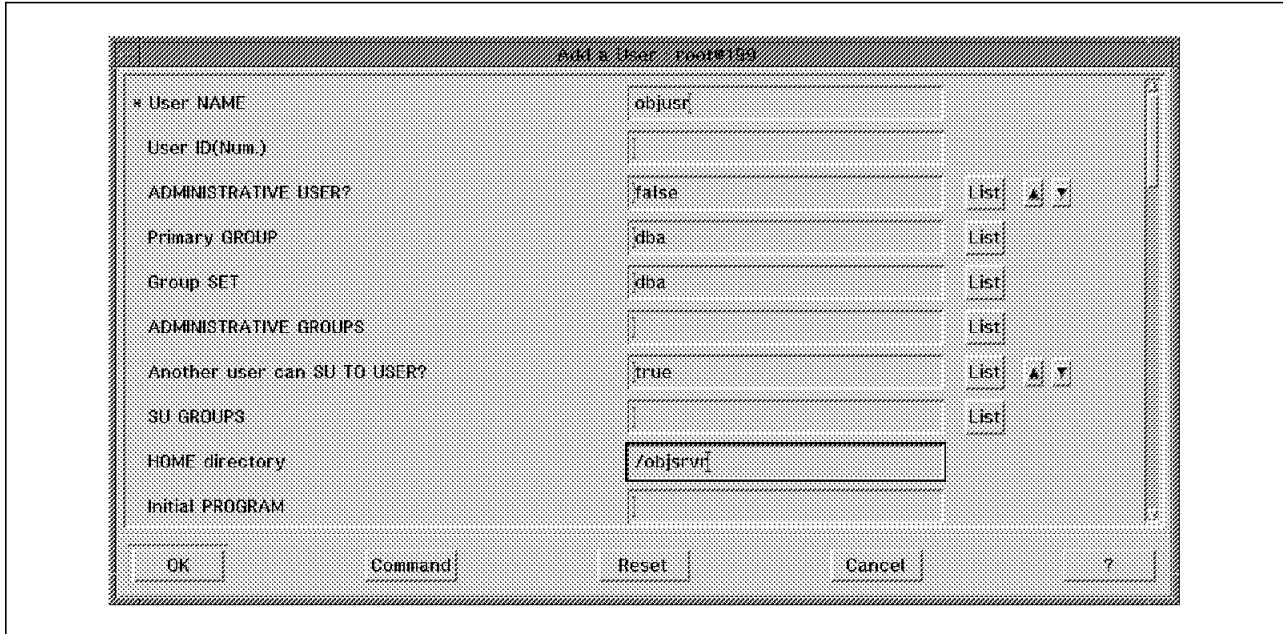
Figure 62. Create an Object Server User ID

Table 10. User ID				
UID	**Primary**	**Group Set**	**Administrative Group**	**Home Directory**
objusr	dba	dba	dba, system	/objsrvr

A.5 Installation of Any Software

Here we describe installation of software in AIX through **smit install_latest** screen. The steps are these:

1. Log on as user root.

2. Type smit at the command prompt.

3. Open **Software Installation and Maintenance** from the list of choices on the System Management Panel.

4. Select **Install and Update Software**.

5. Select **Install/Update Selectable Software(Custom Install)**.

6. Select **Install Software Products at Latest Level**.

7. Select **Install New Software Products at Latest Level**.

8. Click on **List** to select the device or directory from the list of available sources, or specify the input device or directory that is the source of the software you are installing. The input device can be a tape, file system, CDROM, diskette drive, or a directory.

   Then click on **OK**.

9. Click on **List** to select the software components to be installed in the **Software to Install** field. Select the options.

10. Click on **OK** on the confirmation dialog box to apply fixes.

11. Click on **DONE** when the installation program completes successfully.

12. Close SMIT.

A.6 How to Install Fixes

To install the FixPak, do the following:

1. Log on as user root.

2. Shut down all Digital Library components to be updated.

3. After shutting down any running components, and before installing the service pack, issue the command:

   # slibclean

   This will remove any copies of the existing shared libraries from system memory. If you do not have the authority to issue this command, you must log on as *root* to perform this action.

4. Type smit at the command prompt.

5. Open **Software Installation and Maintenance** from the list of choices on the System Management Panel.

6. Select **Install and Update Software**.

7. Select **Install/Update Selectable Software (Custom Install)**.

8. Select **Install Software Products at Latest Level** (Figure 63 on page 86).

9. Select **Install New Software Products at Latest Level**.

10. Click on **List** to select the device or directory from the list of available sources, or specify the input device or directory that is the source of the software you are installing. The input device can be a tape, file system, CDROM, diskette drive, or a directory.

    Then click on **OK**.

11. Click on **List** to select the software components to be installed in the **Software to Install** field. Select the options.

12. Click on **OK** on the confirmation dialog box to apply fixes.

13. Click on **DONE** when the installation program completes successfully.

14. Close SMIT.
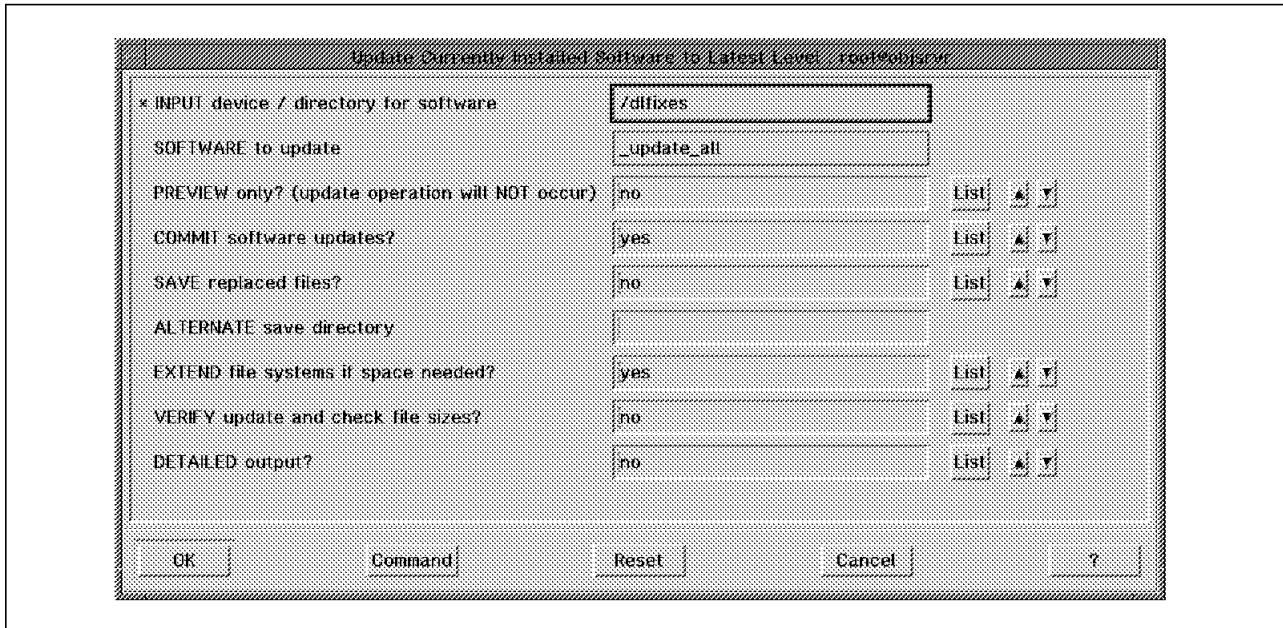
Figure 63. Update Currently Installed Software to Latest Level

Appendix B. Installation of DB2/6000

Digital Library uses DB2/6000 as the back end for storing data and querying data. DB2/6000 has to be installed in both the Library Server and the Object Server machines. Also, two instances have to be created for the Library Server as well as Object Server. Here we describe how to install the DB2/6000 and create instances.

B.1.1 Installing DB2/6000

The steps are these:

1. Insert the DB/2 CDROM in the machine.
2. As *root* user, go to the screen **smit install_latest**. (See A.5, "Installation of Any Software" on page 84).
3. From the list of components to be installed, select the following:
   - DB2 Products Bundle.
   - DB2 C language Include files.
   - DB2 Call level Interface samples.
   - DB2 Client Application Enabler.
   - DB2 Code Page Conversions.
   - DB2 Command Line Processor.
   - DB2 Communications Support-Base with TCP/IP.
   - DB2 Database Director.
   - DB2 Executables.
   - DB2 Performance Monitor.
   - DB2 SDK Utilities and Samples.
   - DB2 Utilities and Samples.
   - DB2 Visual Explain.
   - DDCS DRDA Application Requestor.
4. Click on **OK** to install.
5. Click on **Done** to exit SMIT. The DB2 Software gets installed in the /usr/lpp directory.
6. Change the directory to the following:

   ```
   # cd /usr/lpp/db2_02_01/cfg
   ```

7. Run the script db2ln. This creates links from /usr/lpp/db2/lib to /usr/lib and from the /usr/lpp/db2/include to /usr/include.
8. Change directory to **/usr/lpp/db2_02_01/instance**.
9. Run the following command.

   ```
   # ./db2icrt <instance name>
                  (eg. ./db2icrt libusr)
   ```

   This would create a directory with the name sqllib in the HOME directory of the instance user.

   Note: There should be a user with the name of the instance, before the instance is created.
10. Add the following in the /etc/services file.

   ```
   db2inst1c      3700/tcp        #DB2 Connection Service Port.
   db2inst1i      3701/tcp         #DB2 interrupt connection service port.
   ```

11. Login as the instance user.
12. Copy the file *db2profile* from the directory $HOME/sqllib to the HOME directory of the user as the *.profile* file.

13. Edit the .profile and add the word TCP/IP against the field DB2COMM in the *.profile* file. Uncomment this line and the next (*export*) line. Execute the *.profile*.
14. Type db2 in the command. (It should give some messages and come to a db2 prompt.)
15. Type the following:

    db2 => update database manager configuration using SVCENAME db2inst1c

16. Type quit at the prompt.

Now the DB2 has been installed and an instance created.

B.1.2 Starting DB2/6000.

To start the DB2 instance, login as the instance user and type in the following command:

    $ db2start

This starts the db2 server and other components. The command **ps -fu<username>** should show the db2 processes running.

B.1.3 Stopping DB2/6000.

To stop the DB2 instance, log on as the instance user and run the following command:

    $ db2stop

This stops the DB2 instance for that user.

Appendix C. Special Notices

This publication is intended to help system administrators who have the responsibility to install, configure, and verify a Digital Library Version 1 environment. This publication will also assist Digital Library application development personnel who need to customize a solution with a different data model. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM Digital Library Version 1. See the PUBLICATIONS section of the IBM Programming Announcement for IBM Digital Library Version 1 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 use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX®	DATABASE 2
DB2®	IBM®
OS/2®	

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 D. Related Publications

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

D.1 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

D.2 Other Publications

These publications are also relevant as further information sources:

- *IBM Digital Library - Planning and Installation Guide,* GC26-8623-01

- *IBM Digital Library - Managing Digital Library with System Administration and Database Utilities,* GC26-8626-00

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:

- **PUBORDER** — to order hardcopies in United States

- **GOPHER link to the Internet** - type `GOPHER.WTSCPOK.ITSO.IBM.COM`

- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

      TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
      TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)

  To get BookManager BOOKs of redbooks, type the following command:

      TOOLCAT REDBOOKS

  To get lists of redbooks, type one of the following commands:

      TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
      TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE

  To register for information on workshops, residencies, and redbooks, type the following command:

      TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996

  For a list of product area specialists in the ITSO: type the following command:

      TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE

- **Redbooks Web Site on the World Wide Web**

  `http://w3.itso.ibm.com/redbooks`

- **IBM Direct Publications Catalog on the World Wide Web**

  `http://www.elink.ibmlink.ibm.com/pbl/pbl`

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**

- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to `announce@webster.ibmlink.ibm.com` with the keyword `subscribe` in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

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.htm`). 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	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerobject serverd
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **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

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **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

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

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.htm). 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)

List of Abbreviations

ADSM	automated data storage management	**HTTP**	Hypertext Transmission Protocol
AIX	advanced interactive executive	**IBM**	International Business Machines Corporation
API	application program interface	**ITSO**	International Technical Support Organization
APPC	advanced program-to-program communications	**LAN**	local area network
		MPEG	Moving Pictures Expert Group
ASCII	American national Standard Code for Information Interchange	**OS/2**	operating system/2
		PC	personal computer
ASPI	advanced SCSI programming interface	**PERL**	practical extraction and reporting language
Cliette	internet connection server for AIX	**SCSI**	small computer system interface
CM	connection manager	**SM**	search manager
CSD	corrective service diskette	**SMIT**	System Management Interface Tool
DL	digital library		
DNS	domain name service	**SNA**	systems network architecture
EUIS	end user interface services	**SMS**	system managed storage
GIF	graphical image format	**TCP/IP**	transmission control protocol internet protocol
HTML	Hypertext Markup Language	**VM**	virtual machine

Index

A

abbreviations 121
acronyms 121
AIX
    any software installation 84
    creating groups 81
    creating user IDs 81
    DB2/6000 87
    fixes, installation of 85
    root setting changes 82
    System Management Interface Tool (SMIT) 77
    tailoring 77

B

bibliography 91

C

client
    sample system 1
    troubleshooting tips 39
Cliette
    configuration 33
    installation 29, 33
    sample system 1
codepage
    setting 3, 11
    setting in AIX 77
custom solution
    creating 4
    customizing Internet connection 72
    data modeling 69
    data modeling - control file 70
    data modeling - index class 69
    data modeling - key fields 69
    data modeling - loader script 70
    data modeling - loading data 71
    overview 67
    prerequisites 68

D

data modeling
    control file 70
    index class 69
    key fields 69
    loader script 70
DB2/6000
    installation 87
    starting 88
    stopping 88
Digital Library administration
    troubleshooting tips 39

Digital Library administrator
    installation
        sample system 1
Digital Library client
    hardware 52
    installation 51
    installation steps 54
    network table generation 60
    software 52
    utilities 60
Digital Library utilities
    installation 7
    using 12

F

file system
    adding file system on AIX 79
    creating on AIX 78
    logical volume on AIX 78
    mounting file system on AIX 80
fixes
    corrective service diskettes (CSD) 2
    object server 42

G

generating
    network table 16
glossary 97

H

HTTP
    installation 3, 29, 34
    server 3

I

installation verification
    utilities 60

L

library information
    statistical information 33
library server
    configuration 18
    generating database 31
    generating network table file 31
    installation 7, 29
    sample system 1
    setup file 31
    starting 32
    stopping 33

library server *(continued)*
    troubleshooting tips   39
logical volume
    adding on AIX   78

N
network table
    generating   16

O
object server
    configuration   18, 43
    creating database   43
    DB2 database creation   45
    DB2 environment variables   44
    DB2 instances   44
    fix   42
    generating network table   45
    generating setup file   43
    installation   41
    sample system   1
    starting object server   46
    stopping object server   47
overview
    Digital Library   1
    sample system   1

P
PERL
    installation   3, 29, 34

R
requirements
    hardware   2
    software   2

S
sample data
    installation   29
    loading   26, 38
    viewing   38
sample system
    configuration   2
search manager
    configuration   20, 34
    defining classes   7
    defining indexes   7
    installation   29, 34
    key fields   23
    starting   37
    stopping   37
system configuration file
    creating   12
    defining nodes   14

system configuration file *(continued)*
    generating network table   16

T
testing
    sample system   26

V
VisualInfo
    capturing documents   50, 62
    creating documents   50, 62
    indexing documents   50
    overview   49
    processing documents   51
    searching documents   50, 63
    viewing documents   51, 63

ITSO Redbook Evaluation

Getting Started with Digital Library V1.0
SG24-2243-00

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.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@vnet.ibm.com

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!)**

_____

_____

_____

_____

_____

125

IBM ®

Printed in U.S.A.