ImagePlus VisualInfo Client/Server Solution: Sample Code for Client

Document Number GG24-4369-00

February 1995

International Technical Support Organization
Bethesda Center

Take Note!

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

First Edition (February 1995)

This edition applies to Release 1.0 of VisualInfo and includes the following: Library Server 5655-036, Object Server 5622-213, and Client 5621-326, for use with OS/2.

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

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

IBM Corporation, International Technical Support Organization
Dept. FGC Location BME
6710 Rockledge Drive
Bethesda, Maryland 20817-0000

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

Abstract

This document provides detailed coverage of how to use the VisualInfo application programming interfaces (APIs). VisualInfo differs from previous ImagePlus products in that the APIs run on the image workstation (client) and not on the host. This document provides examples and descriptions of how to implement key image functions such as scan, display, print, import, and export, and includes source code. It provides a quick start for developers of VisualInfo-based applications.

This document was written for application developers and programmers. Some knowledge of image concepts, C language, and PM programming is assumed.

(190 pages)

Contents

Figures

Special Notices

This publication is intended to help application developers and C programmers to develop image applications for the VisualInfo client. The information in this publication is not intended as the specification of any programming interfaces that are provided by VisualInfo. See the PUBLICATIONS section of the IBM Programming Announcement for VisualInfo for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, CT 06904 USA.

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

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

Application Development	DB2/2
IBM	Operating System/2
OS/2	Personal System/2

PS/2 VisualInfo

Preface

This document is intended to assist application developers to use the VisualInfo APIs to create a complete image application, or to provide additional functions to complement the client application. It contains sample programs written by residents. The programs perform operations such as scan, display, and print. These functions are required as a basis for any operational image application.

This document is intended for C programmers and developers.

How This Document is Organized

The document is organized as follows:

- Chapter 1, “Introduction”
- Chapter 2, “The Search Function”

  This chapter describes a sample program that illustrates some of the VisualInfo facilities that you might use for searching the library for a particular document or set of documents.
- Chapter 3, “The Scan Function”

  This chapter describes a sample program that illustrates some of the VisualInfo facilities that you might use for performing a basic scan of a document.
- Chapter 4, “The Import Function”

  This chapter describes a sample program (IP3IMPRT) that illustrates the basic VisualInfo facilities that you might use for importing a PC file into a VisualInfo library.
- Chapter 5, “The Display Function”

  This chapter describes a sample program that illustrates some of the VisualInfo facilities that you might use to display a document from the VisualInfo library. It includes Design Objectives and a description of the Program Logic with references to the API calls used. Points of interest raised during the development of the sample are also covered.
- Chapter 6, “The Print Function”

  This chapter describes the IP3PRINT function that we developed for the VisualInfo environment. It includes Design Objectives and a description of the Program Logic with references to the API calls used. Points of interest raised during the development are also covered in hints and tips.
- Chapter 7, “Create a Printer Option Profile”

  This chapter describes how to create a printer option profile (POP) for the VisualInfo environment. The printer option profile must be created before you can use the print function.
- Chapter 8, “The Export Function”

  This chapter describes a sample program that illustrates some of the VisualInfo facilities that you might use for exporting an object from a VisualInfo library to a workstation file.

- Appendix A, “IP3SRCHD Source Code” through Appendix G, “IP3EXPRT Source Code” contain the source code for the functions described in Chapter 2 through Chapter 8.

Related Publications

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

- *VisualInfo Application Programming Guide, Volume 1*, SC31-7662
- *VisualInfo Application Programming Guide, Volume 2*, SC31-7682
- *VisualInfo Application Programming Reference, Volume 1*, SC31-7663
- *VisualInfo Application Programming Reference, Volume 2*, SC31-7664
- *VisualInfo Application Programming Reference, Volume 3*, SC31-7665
- *VisualInfo Application Programming Reference, Volume 4*, SC31-7667
- *VisualInfo Administration and Operations Guide*, SC31-7661
- *VisualInfo User's Guide*, SC31-7670
- *Real-World Programming for OS/2 2.1*, SR28-4943

International Technical Support Organization Publications

- *Image Processing: ImagePlus and VisualInfo*, GG24-4109

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

Bibliography of International Technical Support Organization Technical Bulletins, GG24-3070.

To get listings of ITSO technical bulletins (redbooks) online, VNET users may type:

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

How to Order ITSO Technical Bulletins (Redbooks)

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

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

Acknowledgments

The project leader was:

John Reid
International Technical Support Organization, Bethesda Center

The authors of this document are:

Christine Gassmann
IBM Switzerland

Willy Stanzl
IBM Vienna

David Thrum
IBM Australia

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

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

Carol Liscinsky
IBM Bethesda Programming Labs

Trina Norden
IBM Bethesda Programming Labs

Namita Singh
IBM Bethesda Programming Labs

Chapter 1. Introduction

The purpose of this document is to provide a set of samples that can be used as the basis for a more complex application that will run on the VisualInfo client. We do not wish to claim originality for all the code provided herein, because we borrowed large portions from samples provided by Bethesda Programming Labs (BPL), Image Services development. We are indebted to BPL for providing us with these samples and letting us use them in our code. The VisualInfo product is shipped with the following samples in source code:

- FRNOEXFO - create a folder
- FRNOEXDO - create a new document from a bit-map file
- FRNOEXNO - add a note to a document
- FRNOEXSE - search for documents
- FRNOEXRE - replace an object with a new object
- FRNOEXDI - display an object
- FRNOLG - logon building block

Note: The samples listed above may vary from release to release. This list is only provided as a guide. Check the product documentation for a list of samples provided in a particular release.

The samples discussed in this document provide additional functions such as scan, print, import, and export, and, in the case of display and search, offer an alternative sample for performing these functions. The following is a summary of the samples discussed in this document:

- IP3SRCHD - search for items
- IP3SCAN - scan in one or more pages
- IP3IMPRT - import a MODCA document from a PC file
- IP3DISP - display a document
- IP3PRINT - print a document
- IP3POP - create a printer option profile
- IP3EXPRT - export a document to a PC file

Note: The samples described herein have not been formally tested and may contain errors and bugs. They compiled cleanly and performed the tasks they were designed to do during our limited testing and usage. You are advised to conduct thorough performance and function testing on any of the code in this book that you use. The programs have limited clean-up and error checking functions built in because we wanted to keep the samples simple. You should include more comprehensive error recovery and cleanup facilities than we did in any production programs you implement.

Chapter 2 through Chapter 8 describe the above functions in detail. One chapter is devoted to each function. The source code files, including the .C, .H, and .MAK files, are included in Appendix A through Appendix G for your reference. They were formatted using the CBOOK tool for ease of cross reference and interpretation.

1.1 Client Setup

We tested this code with the CSD version of VisualInfo dated September 1994. Some packaging changes occurred between pre-GA and GA levels of code. You should note the following:

- If you are developing applications using the APIs on a Stand-alone VisualInfo system, make sure you install the “Application Programming Toolkit” as part of the VisualInfo install otherwise the INCLUDE files and other components may not get installed.
- If you plan to program on a client as we did (with a separate server) be aware that installing the Standard Client does not include the “Application Programming Toolkit” option and it cannot be selectively added later. You must therefore install the client from the “Selectable Components” install option. We selected the following options to install with the client:
  - Library Client
  - Library Client Programming Toolkit
  - Client Application Programming Toolkit
  - IBM Image Services Toolkit
  - Client Application
  - IBM Image Services

1.2 Sample Diskette and Compiling Programs

You will find a diskette in the back of this document which contains the sample code described in this book.

You can install the sample code on your workstation using the following commands:

1. Make a directory on your C: or D: drive called “EXAMPLES” (MD EXAMPLES).
2. Change directory to “EXAMPLES” (CD EXAMPLES).
3. Place the examples diskette in the A: drive and issue the following command: XCOPY A:*.* /S

This will create separate subdirectories for each sample function.

Compile the programs by first changing to the appropriate subdirectory (for example, CD IP3SRCH), then issue the appropriate MAKE command (for example, NMAKE IP3SRCHD.MAK).

Execute the programs and test each individually. If you have any problems check that the .H file and parameters are correct for your system.

Note: It is recommended that you make sure your workstation operates correctly with the VisualInfo Client Application before testing these programs.

Chapter 2. The Search Function

This chapter describes a sample program that illustrates some of the VisualInfo facilities that you might use for searching the library for a particular document or set of documents.

2.1 Objectives

The overall objective of the IP3SRCHD function is to find particular documents given certain search criteria. The search criteria are specified as attributes within an Index Class. For example “Last name” could be an attribute within the Index Class of “Claim”. Index Class and Attribute (with value) are the minimum criteria for a meaningful search. For more information about Index Classes see *VisualInfo Administration and Operation Guide*.

The IP3SRCHD function was designed to perform the following steps:

- Establish Folder Manager session
- Get list of Index Class IDs
- Get Index Class information
- Get list of views
- Get list of attributes
- Search for items and store result
- Terminate Folder Manager session

Note: This program has been designed to show the internal operations in a search. It produces a large amount of output which you would suppress in a production program.

2.2 Program Design

To search for an item, the *SimLibSearch* API is used. (See Appendix A, “IP3SRCHD Source Code” on page 57 for a listing of this program.) To supply this function with the necessary information, the program has to know about the internal representation of the descriptors of an item. Items can be supplied with *attributes*. One or more attributes make up an *Index Class*. An item can be assigned an Index Class. In other words, the Index Class is the set of attribute fields which are used to uniquely identify an item. Views are used to narrow the set of attributes eligible to a user for selection. Views are a logical security function. Views have an influence on the search capabilities of a user or program. One item can have just one Index Class but one Index Class can have many views.

There is a difference between the external presentation of attributes to a user, and the internal representation that a program uses to request services from VisualInfo. A user uses meaningful names, while most VisualInfo functions internally use attribute identifiers which are numbers. The same is true for Index Classes. To translate user-friendly names to program-friendly numbers one has to access the tables which describe the attributes and the Index Classes.

The search function returns item IDs which can be used in further processing to access the items. These item IDs are written to the file specified in the /F parameter and to the console.

The following is an example of an item ID:

```
Y6156789J6788226
```

Note: The program produces a fair amount of output prior to the item ID. This output is useful for debugging, understanding the internal workings of the program, and demonstrating how VisualInfo implements the search function. In an actual production search program you would suppress this extraneous output.

The program must log on to the Folder Manager prior to performing the search, and log off when the search has been completed.

The sample program IP3SRCHD.EXE performs these functions. IP3SRCHD accepts the following command line parameters:

- /I=IndexClass (for example, Claim)
- /AAttribute=Value (for example, “Last name=Jones”)
- /F=OutFileName

The user ID and password used to log on to the Folder Manager are hard-coded in the program.

The /I parameter is reserved for future use to specify Index Class. IP3SRCHD checks for this value but does not use it.

The following example shows how you might start IP3SRCHD:

```
[C:FRNV1ROEXAMPLESIP3SRCH]
      ip3srchd /F=ITEMS.RES /Aamount>100000 /Acusto=FRX459 /I=Comm
```

If the attribute name contained a blank you could enter the /A parameter as follows:

```
[C:FRNV1ROEXAMPLESIP3SRCH]
      ip3srchd /A"Last name=Jones" /F=out.fil /I=Claim
```

2.2.1 Program Flow

The IP3SRCHD program flow is as follows:

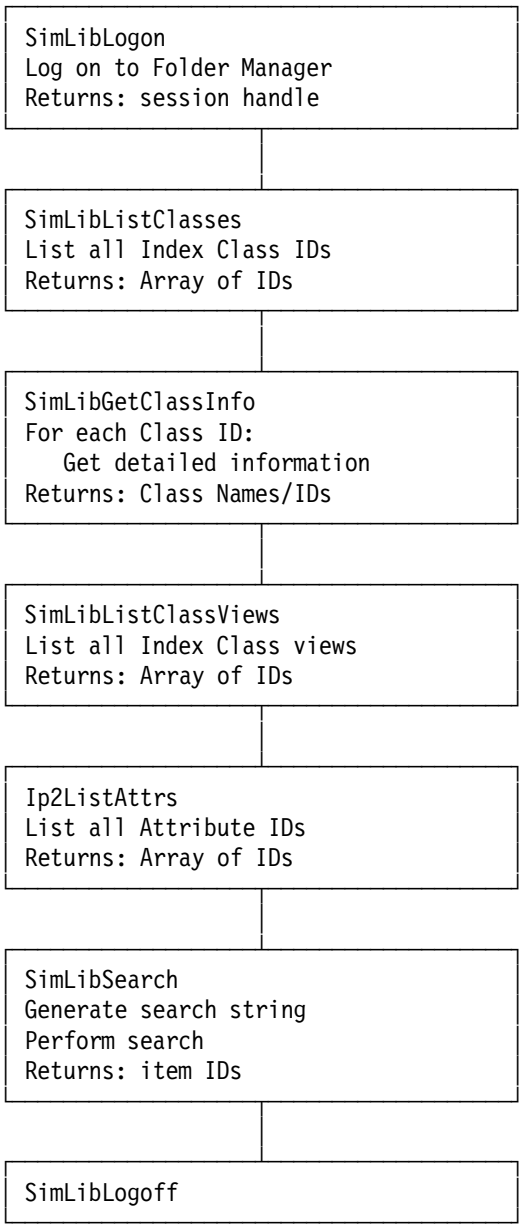


Figure 1. Program Flow for IP3SRCHD

2.2.2 APIs Used

SimLibLogon

A VisualInfo Folder Manager API

You must log on to the library server to obtain a session handle (hSession) before issuing any other VisualInfo APIs. The SimLibLogon API is used to establish a session with the Folder Manager. The initial values for this API are taken from the .H file (IP3SRCHD.H), where they are hard-coded. It would be just as easy to pass them to the main program via command line parameters.

SimLibListClasses

A VisualInfo Folder Manager API

Lists all Index Class IDs which have been defined. The IDs are numbers. To associate them with the names from the program parameter /I, a SimLibGetClassInfo call is issued next.

The parameters necessary to run this function are the session handle (hSession) of the current Folder Manager session as returned from SimLibLogon, a class options parameter, a processing option parameter and a pointer (pRC) to a return code data structure RCStruct.

The class options parameter defines which classes shall be listed by SimLibListClasses.

The processing option parameter defines whether this request has to be done asynchronously from the main program or whether the program has to wait for the completion of this functions. Specifying the NULL pointer, as in our example program, means synchronous processing, so we wait until the process completes.

The last parameter is the parameter for the return code data structure RCStruct, which gives more information about the return status of this function. It also contains a field which is API dependent. In this case it is a pointer to an array of structure NAMESTRUCT which contains IDs and additional information. One field (ulParam2) returns the number of elements in that array, and ulParam1 is a pointer to the array. Do not be concerned about the name field Namestruct.szName since it is intended for use by FaxRouter/2 and may contain different class names from the actual ones. So the only information we are using from this array of structures is the list of classes that are defined, and their numerical IDs.

SimLibGetClassInfo

A VisualInfo Folder Manager API

To get the real name of an Index Class, the SimLibGetClassInfo API has to be invoked for each Index Class ID. SimLibListClassInfo behaves in a similar way to SimLibListClasses. SimLibListClasses can list details for Index Classes or for Index Class views. The actual input to SimLibGetClassInfo is the class type (here SIM_INDEXCLASSID indicate that an Index Class ID follows) and an Index Class ID. Again, besides a return code, the return code data structure RCStruct contains a pointer to a data structure of type CLASSINFOSTRUCT (see *VisualInfo Application Programming Reference, Volume 3* for detailed descriptions of data structures). This data structure contains the name of the specified Index Class.

SimLibListClassViews

A VisualInfo Folder Manager API

This function works in the same way as SimLibListClasses, except that it returns Index Class Views rather than Index Classes. Note that every Index Class has at least one view, namely itself.

Ip2ListAttrs

A VisualInfo Folder Manager API

Similar to SimLibListClasses, this function returns all attributes known to the VisualInfo database. Ip2ListAttrs does not take any Index Classes or Index Class views into consideration. It simply returns the numeric IDs of all attributes it knows.

Note that every VisualInfo installation has a basic set of Index Classes and attributes that are created when the system is installed. They are for system use, but still show up in the list API results. So, to try out the IP3SRCHD program, you need not define your own Index Classes or attributes. IP3SRCHD will list the predefined classes and attributes.

SimLibSearch

A VisualInfo Folder Manager API

At this point the program knows the following information:

- All Index Class IDs and names
- All index view IDs
- All attribute IDs and names

User input that is used to create search argument(s) for SimLibSearch are:

```
Attribute names          ( from /A parameter)
Attribute value          ( from /A parameter)
Comparison operator      ( from /A parameter)
```

The application program must now translate all names into numerical IDs preceded by the character “A”, translate the symbolic operators **<**,**>**,**!**, and **=** into their corresponding values LT, GT, NE, and EQ. You may have more than one comparison expression by simply specifying multiple /A parameters. If there are more than one they are connected by AND operators.

For example, the following call:

```
[C:FRNV1ROEXAMPLESIP3SRCHD]
IP3SRCHD /F=ITEMS.RES /Aamount>100000 /Acusto=FRX459 /I=Comm
```

could produce the following search argument string:

```
A075 GT ″100000″ AND A107 EQ ″FRX459″
```

Note that numerics in quotes have the same effect as ones without quotes.

IP3SRCHD translates and passes all four types of comparison operators to SimLibScan.

More than one search string can be passed to SimLibSearch in a single call. The search criteria is passed as an array of structures of type LIBSEARCHCRITERIASTRUCT, together with the number of elements in it.

The call to SimLibSearch in IP3SRCHD uses the following parameters:

Parameter 1 is the session handle (hSession) to the Folder Manager session opened by SimLibLogon.

Parameters 2 and 3 are additional filters to limit the scope of the search process and to special link criteria. These two parameters are not used by the current release of VisualInfo.

Parameter 4 specifies the way SQL shall process the search request. There are the following choices:

```
SIM_SEARCH_BUILD_ONLY      build static search, no actual selection
SIM_SEARCH_STATIC          build static search, do selection
SIM_SEARCH_DYNAMIC         use dynamic SQL
SIM_SEARCH_STATIC_ONLY     use previously built and stored request
```

Parameter 4 influences performance significantly. Every search request is passed to the OS/2 Database Manager as an SQL query. SQL queries can exist in two different forms: static or dynamic. Dynamic SQL is very flexible and does not need a bind of a query into the program, but has to be interpreted by the SQL optimizer before execution. Static SQL is a prepared and bound form of SQL statement ready for immediate execution.

A single dynamic SQL query may be better than building a static one if the query is only to be executed once. However, if many calls of the same SQL request are to be performed (arguments can be changed, but not the row names and operators), static SQL will outperform dynamic SQL. IP3SRCHD uses SIM_SEARCH_STATIC, which specifies that if a static query exists it should be used; otherwise, use a dynamic call. All SQL queries are stored by VisualInfo and a background process converts them to a static query once a threshold has been reached.

Parameter 5 specifies the item type to search for:

```
SIM_DOCUMENT               return document IDs
SIM_FOLDER                 return folder IDs
SIM_FOLDER_DOC             return folder IDs and document IDs
```

IP3SRCHD is designed to find only documents, so we specify SIM_DOCUMENT.

Parameter 6 specifies the item location regarding workflows. It can be any combination of the following values:

```
OIM_ITEMS_NOT_IN_WORKFLOW      no workflow
OIM_CURRENT_WORKFLOW_ITEMS     in any workflow
OIM_CANCELLED_WORKFLOW_ITEMS   removed from workflow
OIM_COMPLETED_WORKFLOW_ITEMS   completed in a workflow
OIM_ALL                        don't care.
```

You can combine the values using the C language bitwise OR operator "|".

Parameter 7 specifies the item status. It can be any combination of the following values:

```
OIM_ITEMS_SUSPENDED            suspended items
OIM_ITEMS_NOT_SUSPENDED        items not suspended
OIM_ALL                        don't care
```

Once again you can combine the values using the C language bitwise OR operator "|".

Parameter 8 is necessary when a folder is to be created (see parameter 11) to hold the search results. We request the search results be placed in a buffer, so this value is ignored and we set it to zero. Do not confuse this with a search argument. SimLibSearch does not take Index Classes into consideration. It simply processes attributes. The /I parameter passed to IP3SRCHD is checked but not actually used by the program.

Parameter 9 is the number of search criteria array elements. In IP3SRCHD it is always 1 because all search arguments are concatenated by ANDs to one single search argument string.

Parameter 10 is a pointer to an array of structures of type LIBSEARCHCRITERIASTRUCT. See above for a detailed description. This contains a pointer to the search criteria string.

Parameter 11 defines whether the search result (document or folder IDs) shall be put into a folder or into a buffer (memory). Use of a buffer is requested by specifying the Boolean value TRUE.

Parameter 12 sets the processing mode (synchronous or asynchronous, see above).

Parameter 13 is a pointer to the return code data structure RCStruct which also returns a pointer to the buffer holding a list of the IDs of the item IDs found, and the number of IDs returned.

Now IP3SRCHD reads the item IDs out of the buffer and writes them to the file specified by parameter /F and to the console with a printf. To find the file, the OS/2 function DosSearchPath searches the file name in OS/2s DPATH, much as an AS/400 would do with its library list. If the file is not found, it will be created in the current directory.

SimLibLogoff

A VisualInfo Folder Manager API

Terminates the session with the Folder Manager.

2.3 Implementation Hints and Tips

IP3SRCHD does no SimLibFree calls to free memory used by the API calls. Being a single-threaded program which was designed to be run as a command from the command line and then terminate (and therefore release all storage belonging to it) it does not need to make these calls. It would be better programming practice to use SimLibFree as soon as the returned information from the APIs (such as SimLibListClasses and so on) is no longer needed. SimLibFree only requires the session handle and a pointer to the buffer you are freeing up (and a return code structure) so it is fairly easy to use.

If you use parts of SimLibSearch repeatedly, for example in an application that processes multiple requests, it might improve performance to keep the tables holding all IDs and names in storage instead of retrieving all attribute information each time it is needed. In this case you would not do SimLibFree calls, but rather save the pointers to the table locations and reuse the ID and names tables to improve application performance.

Chapter 3. The Scan Function

This chapter describes a sample program that illustrates some of the VisualInfo facilities that you might use for performing a basic scan of a document.

3.1 Objectives

The objective of the IP3SCAN sample is to scan one or more pages into VisualInfo and to store them as an item part. To perform this function we must do the following steps:

- Initialize Presentation Manager because the scan function must have a parent window for control and termination (WinInitialize)
- Create a PM Message Queue and a window
- Establish a session with the Folder Manager
- Create a display window to display the scanned image (WinCreateStdWindow)
- Start a thread to perform the scan (DosCreateThread)
  - Initialize PM environment for this thread (WinInitialize)
  - Scan the document into the Working Set (SimScnBasicScan)
  - Query WS to check that a document has been scanned (SimWsQueryObject)
  - Create a new document item to store the scanned document (SimLibCreateItem)
  - Create a new object in this document to contain the scanned document (SimLibCreateObject)
  - Free the working set (SimLibFree)
  - Store the document from WS into this library object (SimWsStoreObject)
- Process the message loop to wait for user to quit (WinGetMsg)
- Logoff from the Folder Manager session (SimLibLogoff)
- Exit main() with a return code

3.2 Program Design

To scan an image, access to various VisualInfo functions must first be established. A *Folder Manager* session must be started before any further API processing can be done and so that the scanned object can be stored in the library.

The *Image Services* environment must be initialized, and a *Working Set* object created (SimWsCreateObj) to contain the scanned image. A display window must be created to display the contents of this working set object, and scanning services started to scan the document into this display window and working set object. Also, since both display and scanning services are Presentation Manager functions, the scanning program must be a Presentation Manager application, and initialized accordingly.

After a successful scan, the resulting object must be stored from the working set into the VisualInfo library.

Finally, image services cleanup, (see function “Cleanup” in the source code) PM cleanup and logoff from the Folder Manager should be performed.

Program IP3SCAN performs these functions.

To call the scan function use the IP3SCAN command. IP3SCAN is an EXE file which accepts the following parameters:

- /U=User ID
- /P=Password
- /S=Server
- /N=Name of Application

The */U, /P, /S* and */N* parameters support information for the Folder Manager logon procedure. They have default values which are defined in the include file IP3SCAN.H.

This is an example for a call of IP3SCAN:

```
[C:FRNV1ROEXAMPLESIP3SCAN] ip3scan /U=frnadmin /P=password
```

You can simply use the defaults in the header file for /U and /P and simply issue the command IP3SCAN if the defaults are set up right.

3.2.1 Program Flow

The IP3SCAN program schematic is shown below:

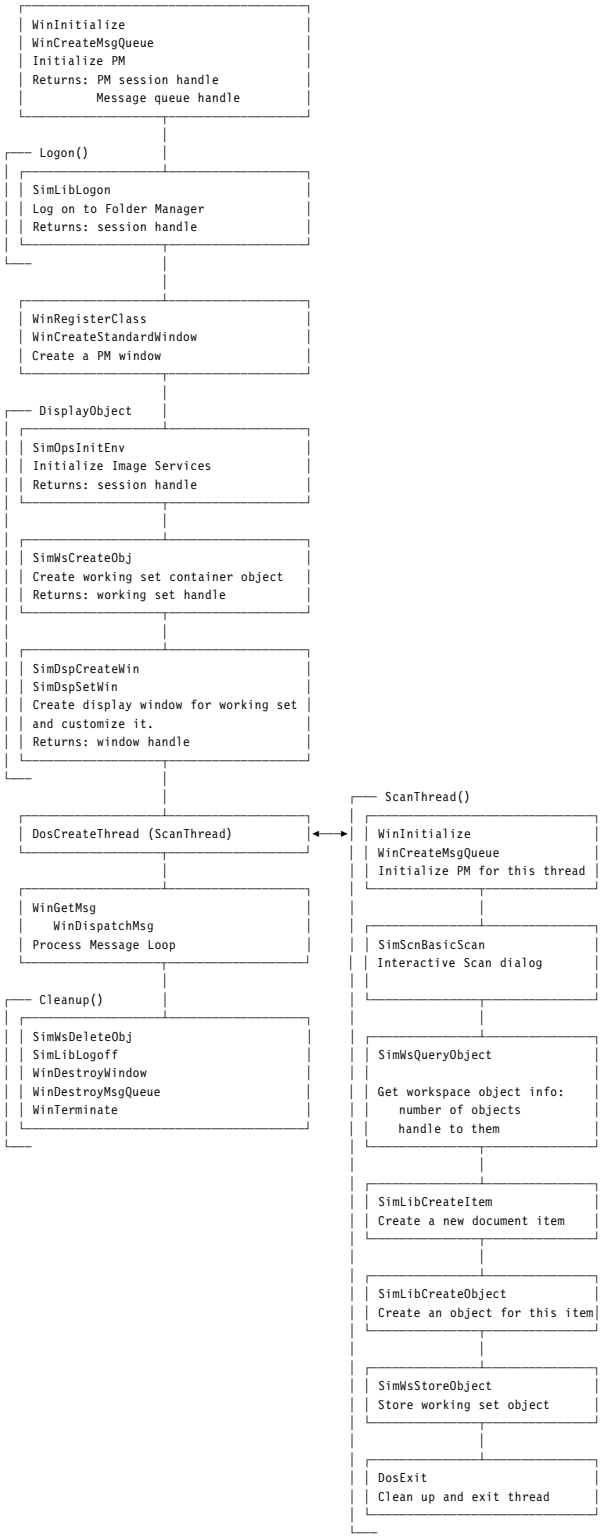


Figure 2. Program Flow for IP3SCAN

WinInitialize

Provided by OS/2 Presentation Manager

This API initializes the OS/2 Presentation Manager environment and returns the PM anchor block handle to this environment. Other PM functions require this anchor block handle. We need to perform this function because VisualInfo creates windows in functions that follow.

WinCreateMsgQueue

Provided by OS/2 Presentation Manager

This API creates an OS/2 Presentation Manager message queue. Windows talk to each other and to their associated Window Procedure through message queues. The messages in the message queue are processed by the application or by the OS/2 Presentation Manager.

3.2.2 APIs Used

SimLibLogon

A Folder Manager API

SimLibLogon establishes the session to the Folder Manager. Uses the parameters /U, /P, /S and /N or their default values as defined in IP3SCAN.H.

WinRegisterClass and WinCreateStdWindow

Provided by OS/2 Presentation Manager

Create a PM Window and specify its Window Procedure. We need to display a PM window to provide the user with a mechanism to terminate IP3SCAN once the document has been scanned and stored in the VisualInfo library.

SimOpsInitEnv

A VisualInfo Image Services API

This API initializes the Image Services environment for our program.

The first parameter is any descriptive name we choose to provide for our application. A NULL pointer would be used if no description is provided.

Setting the second parameter to FALSE means attended mode, so informational messages will not be suppressed.

The third parameter which specifies a working directory for temporary files is set to D:FRNV1R0. You could change it to a path suitable for your application or retrieve its value from an environment variable using the OS/2 Control Program function DosScanEnv.

The next parameter, the user exit DLL directory, is specified as ″D:FRNV1R0DLL″. This is where the default user exits provided with VisualInfo reside. For C language string rules a backslash () has to be specified twice to be interpreted correctly.

Parameter 5 is not used so we set it to NULL.

Parameter 6 specifies the location of the Object Index Manager component of the Folder Manager. This path is used by image services APIs to access data on the server, for example SimWsStoreObject. If we simply specify the module name, VisualInfo will find the module using the standard OS/2 library search order. You can also set it to NULL.

The value we used for this parameter is:

```
FRNOFI
```

Parameter 6 specifies the location of the Object Manager modules. Its use is similar to the previous parameter. The value we used for this parameter is NULL.

Be careful when moving applications from one environment to another. If directory structures or disk letters change, you might overlook the hard-coded path names in the programs.

Parameter 7 specifies additional help files you may be using with this application instance. The value we used for this parameter is NULL.

The last parameter is the pointer to the return code data structure RCStruct.

Note that there is no connection to the actual object manager session, so no parameter hSession is required. Nevertheless image services will use object and object index manager functions by calling the respective DLLs directly as specified in parameters five and six.

SimWsCreateObject

A VisualInfo Image Services API

SimWsCreateObject creates a working set or, in other words, a set of work areas to work with image services data. The working set is temporary and will no longer be needed once the objects have been stored in the VisualInfo library. The parameters define the logical location (and therefore the order of appearance in the work windows) of the newly created working set. Although our example scans in a set of objects to a new working set, the API provides the flexibility to store new objects in an existing working set. There is additional control information which we allowed to default to standard values. SimWsCreateObject returns a handle to the working set “object” in parameter 3 and return code information as usual in the return code data structure RCStruct in parameter 4.

SimDspCreateWin

A VisualInfo Image Services API

This API creates a new OS/2 Presentation Manager display window for the working set objects. This relieves us from the task of creating our own display windows using OS/2 Presentation Manager functions. Windows created by SimDspCreateWin can be children of windows our PM dialog created previously. We want the desktop as parent and owner of the new window, so we set parameters 1 and 2 to HWND_DSKTOP. A value of NULL in these parameters will also be taken as desktop.

Parameter 3 is an optional pointer to an icon. We use NULL.

Parameter 4 is a string containing text describing the window, or a NULL pointer if none.

Parameter 5 is the handle to the Working Set object created previously using SimWsCreateObject.

The next three parameters define the form, location and visibility of the window. The last two parameters are handles to the window itself and the return code data structure RCStruct.

Note that SimDspCreateWin returns control as soon as the window has been created. The user can then interact with the created window and working set object independent of other functions that your program might perform.

SimDspSetWin

A VisualInfo Image Services API

This API customizes a previously created display window. SimDspSetWin provides a useful way to configure the display window that is used in conjunction with the working set.

With one or more calls of this API you can put fields into the window to display various information about your working set (for example, the number of pages currently contained) and control fields (for example, paging forward and backwards). You can also specify whether the window currently is closable or not and its location.

SimScnBasicScan

A VisualInfo Image Services API

Scan the document from the scanner into the working set object. The actual scan function is straightforward if you use the basic scan function. SimScnBasicScan refers to all previously set parameters to get its operational values or takes default values. So we just have to invoke the function, provide the handle to the display window we created before, and provide a return code data structure RCStruct. There is one more parameter: the name of the operational scanning window. It is optional.

You should perform a SimScnRegisterScanner prior to this to specify a default scanner if there is more than one scanner defined to your system.

Note also that SimScnBasicScan does not store the scanned data permanently. The scanned image resides in the working set object only and has to be stored into an item by SimWsStoreObject.

Unlike SimDspCreateWin, SimScnBasicScan does not return control to the calling application until the user closes the SCAN dialog box. This is the reason it must be called from a separate OS/2 thread so that other functions in our program (such as the display window) can continue to execute.

SimWsQueryObj

A VisualInfo Image Services API

This function retrieves status information about the objects contained in a working set.

Remember that all information residing in a working set is temporary. To store that information, certain data must be retrieved from the working set. Most important are:

- Number of objects in the working set
- An array of pointers (or handles) to these objects

SimWsQueryObject receives the handle to the working set to be queried, the user type (which influences which objects are visible for that call), and the type of information to receive. It returns the requested information in a data structure and the return code data structure RCStruct

You may use several calls to SimWsQueryObject to get all information you need, although we only do one.

SimLibCreateItem

A VisualInfo Common API

This API creates an item. The item with its item parts is required to hold the scanned information. This API needs the handle to the current VisualInfo session (created by SimLibLogon), the item type to be created (we create a document, so we specify SIM_DOCUMENT). The document will be created with three basic system-defined attributes, USERID, SOURCE and TIMESTAMP, that are defined for the Index Class NOINDEX. The parameters Index Class, number of attributes and attribute list are set to SIM_INDEX_NOINDEX, 3 and a pointer to an array of structures of type ATTRLISTSTRUCT. Access control is set to SIM_ACC_SYSTEM_CHOOSE which lets the system decide which type of access control applies to this item. (The current release of VisualInfo does not support this parameter.). We chose synchronous processing by specifying NULL for AsyncCtlStruct. The final parameter is the return code data structure.

SimLibCreateItem creates an ID internally used by VisualInfo. It is for use only in programs and not a sensible name for an item from a user's perspective. A user defines *attributes* to name an item at a later time. The following is an example of an item ID:

```
Y6156789J6788226
```

SimLibCreateObject

A VisualInfo Folder Manager API

SimLibCreateObject creates an *Object* and optionally an *Item* if you have not created an itemID already. See also *SimLibCreateItem* and *SimLibCatalogObject* in "SimLibCreateItem" on page 23.

This particular call to SimLibCreateObject creates an object of type MO:DCA-P (IS/2), which is associated with the item created earlier using SimLibCreateItem. The object is described by the pointer to its handle block hObj.

This new object serves as a container for our copy of each working set object. Initially, it does not contain any user data.

SimWsStoreObject

A VisualInfo Image Services API

This API stores a single working set object into an item.

We specify one of the pointers to the working set object parts pointer array to identify the data to be stored as parameter 1.

Parameter 2 defines the content class (data format) to be created out of the data residing in the working set object. This parameter should match parameter 3 of the SimLibCreateObject command. Note that the SimWsStoreObject function does not check the content class of the object which will hold the data against the value of parameter 2. The content class stored with the object description is used by other VisualInfo functions to determine how to handle the data correctly.

This program does not support annotations, so we do not request storing them and specify FALSE as parameter value 3 for SimWsStoreObj.

Parameters 4 and 5 describe the source and destination of the data. The *data source* is *server*, meaning object server - working set object. The corresponding parameter value for this destination is SIMSRC_SERVER_OBJ. There are more parameters required for this type of parameter which have to be supplied in a data structure of type SIMSRCSERVEROBJ. It contains:

- The handle to the object manager session
- The handle to the object into which the working set object has to go
- The access mode for the object (see table below)
- The task priority for the object manager

To select a value for access mode, choose one of the following:

```
SIM_ACCESS_SHARED_READ   open object for read
SIM_ACCESS_EXCL_READ     open object for exclusive read
SIM_ACCESS_READ_WRITE    open object for read/write
```

Parameter 6: There is no segmented storage, so we do not need a handle for segmented storage operations and so we specify NULL.

The last parameter is the pointer to the return code data structure RCStruct.

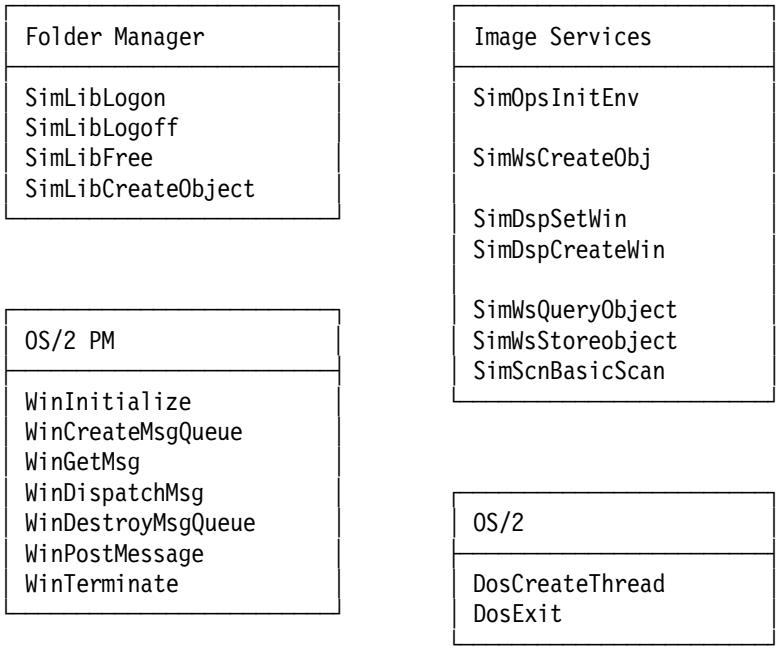
You can use SimLibChangeIndexClass to specify an Index Class for the object, and use SimLibWriteAttr to assign attribute values to that object. (The index class was specified in SimLibCreateItem.) IP3SCAN does not take care of that, however, you may use these functions and consider SIM_OPEN instead of SIM_CLOSE as the Create Control value for SimLibCreateObject to leave it open for those functions.

WinDestroyMsgQueue and WinTerminate

Provided by OS/2 Presentation Manager

This API terminates the PM environment.

3.3 API Type Overview



Refer to *VisualInfo Application Programming Reference, Volume 1* for a detailed description of the Folder Manager APIs. Refer to *VisualInfo Application Programming Reference, Volume 2* for a detailed description of Image Services APIs, and refer to *Real-World Programming for OS/2 2.1* for more information about OS/2 PM calls.

See Appendix B, “IP3SCAN Source Code” on page 73 for a listing of IP3SCAN.C, and the files IP3SCAN.H, IP3SCAN.MAK and IP3SCAN.DEF.

3.4 Implementation Hints and Tips

As in any OS/2 Presentation Manager application, be aware of the fact that you cannot output to STDOUT using the printf() function. Instead, information and error messages are displayed via a PM message box using the WinMessageBox function.

Apart from that, don't forget that a PM program, once started, does not process any user input from the keyboard or the mouse until a *WinGetMessage* has been performed. That means, for slow processes such as *SimLibLogon* and *SimDspCreateWin* the system may seem to be frozen although the mouse pointer still moves and other tasks still update their windows.

Common PM programming rules suggest that not more than 100 milliseconds should elapse until a *WinGetMessage* is issued. To satisfy this PM-specific rule we suggest you expand your program and make long-running functions a separate thread. Threads can be seen as special function calls which, once started, run independently from the program that initiated them. Threads are still part of the main program, however, and can access global variables normally. In the meantime, the main program can enter the message loop to process pending window messages and hence allow the user to do normal windows operations such as changing to another window, moving or resizing the window or requesting the task list.

IP3SCAN was written with the bare minimum of function required to be a "well-behaved" PM program with a message queue and window procedure. A more complete program would provide the user access to facilities via an action bar and pull-down menus from its main window.

See *DosCreateThread* in *Real-World Programming for OS/2 2.1* for information about threads.

3.5 Scanner Setup

To make sure that the scanner was working on the client workstation we first tested it with the Client Application.

We attached the IBM 2456 scanner to the client workstation as follows:

- Checked CONFIG.SYS and added the following statements (if missing):

  ```
  BASEDEV=IBM2SCSI.ADD

  BASEDEV=OS2SCSI.DMD

  DEVICE=D:\FRNV1RO\DLL\EXY24.SYS
  ```

  You can find where the scanner driver (EXY24.SYS) is located by using the OS/2 Seek and Scan Files utility.
- Shut down VisualInfo, OS/2, and powered down the workstation.
- Attached the IBM 2456 scanner to the external SCSI port.
- Powered up the workstation and performed an automatic configuration to add the scanner.
- Tested Basic Scan on the Client Application.
- Tested the IP3SCAN program.

Chapter 4. The Import Function

This chapter describes a sample program (IP3IMPRT) that illustrates the basic VisualInfo facilities that you might use for importing a PC file into a VisualInfo library.

4.1 Objectives

Many image systems obtain their documents and images from an external source in the form of a file. These files can be brought into the VisualInfo system with the import function. See Figure 4 on page 25 for a list of acceptable document types. The objective of this sample is to do a basic import of an object from an OS/2 file.

The IP3IMPRT function was designed to provide the following:

- Establish Folder Manager session (SimLibLogon)
- Create an item (SimLibCreateItem)
- Import a PC file (SimLibCatalogObject)
- Free memory used by the object (SimLibFree)
- Terminate Folder Manager session (SimLibLogoff)

4.2 Program Design

To call the Import function use the IP3IMPRT command.

IP3IMPRT is a .EXE file which accepts the following parameters:

- /U=User ID
- /P=Password
- /S=Server
- /N=Name of Application
- /F=filename
- /I=IndexClass
- /W=Workbasket
- /Afieldname=fieldvalue

The /U, /P, /S and /N parameters support information for the Folder Manager logon procedure. They have default values which are defined in the include file IP3IMPRT.H.

The /I, /W and /A parameters are reserved for future use to specify further item and document information. The current version of IP3IMPRT checks for these values but does not use them.

The /F parameter specifies the OS/2 file name containing the data to be imported. This is a required parameter.

The content class is set to SIM_CC_MODCA_IS2 in this example.

The following is an example of using IP3IMPRT:

```
[C:FRNV1ROEXAMPLESIP3IMPRT] ip3imprt  /F=C:\OBJECTS\CAR1.MDA
```

The above example uses the defaults in the header file for user ID, password and the other parameters.

4.3 Program Flow

IP3IMPRT program flow:

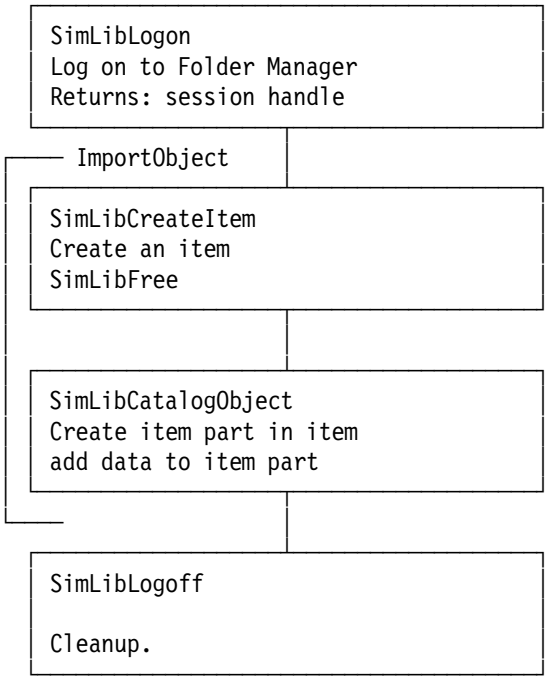


Figure 3. Program Flow for IP3IMPRT

4.3.1 API Description

SimLibLogon and SimLibLogoff

VisualInfo Folder Manager APIs

As you are already familiar with Folder Manager, you know about the *SimLibLogon*, *SimLibLogoff* and *SimLibFree* APIs that were discussed in the earlier chapters. They establish and terminate the Folder Manager session and free the memory used by data structures created during execution of the APIs.

SimLibCreateItem

A VisualInfo Folder Manager API

The smallest piece of information that can be stored by VisualInfo is an *item part*. Usually item parts contain unformatted data such as a scanned image (sometimes called binary large objects or BLOBS), although any kind of data can be stored in an item part. An item part is referred to by an *item part number*.

An item part cannot exist by itself; it needs an *item* as a shell to hold one or more item parts. An item is referred to by an *item ID*. Items, in turn, are stored in *documents*. A document can contain additional information such as *annotations*.

Optionally, documents can be contained in folders and folders can be contained in another folder. Folders cannot be contained in documents.

In other words, there has to be an existing document (item) before you can store information. An item will either be automatically created when you store the data with SimLibCatalogObject or you may provide an existing one. The API to create an item is *SimLibCreateItem*.

Once a folder or document is created, it may be put into a *workbasket* to initiate it in a workflow. This can be done at any time, regardless of whether there are items in the document or not. The normal way of doing this is to supply the data and then place the document or folder in a workbasket. This current version if IP3IMPRT does not add the document to a workbasket.

Note: The following discussion about Item Parts can also apply to APIs such as SimLibCreateObject, SimLibCatalogObject and so on.

Item Parts have an *item type* which is used by other VisualInfo functions to know how to treat the information contained in the item part. This is called the *content class*. *VisualInfo Application Programming Reference, Volume 1* suggests the use of *Ip2ListContentClasses* to get a list of all supported content classes. However, it is easier to check the C header file FRNV1R0INCLUDEFRNPCAPI.H which contains the definitions of all usable content classes in form of C preprocessor definitions. See Figure 4 for VisualInfo document types. Refer to VisualInfo documentation in case these definitions change with releases.

```
/* CONTENT CLASS VALUES */

/* Values from 0-4095 are reserved for IBM definition */
/* Values from 4096-32767 are for application use.    */
/* Values from 32768-65535 are reserved for IBM definition. */

/* IBM "OCA" and TIFF codes */
#define SIM_CC_UNKNOWN        0  /* Content class unknown */
#define SIM_CC_MODCA_IS2      1  /* MODCA-P document (IS/2) */
#define SIM_CC_TIFF5          2  /* TIFF V5, multi-page allowed */
#define SIM_CC_MODCA_PAGE     3  /* MODCA, page structure only */
#define SIM_CC_IOCA_FS11      4  /* IOCA data only (FS11) */
#define SIM_CC_IOCA_TILED     5  /* tiled IOCA only */
#define SIM_CC_GOCA           6  /* GOCA data only  */
#define SIM_CC_AOCA           7  /* AOCA data only  */
#define SIM_CC_BCOCA          8  /* BCOCA data only  */
#define SIM_CC_TIFF5_PAGE    11  /* TIFF V5, single page  */
#define SIM_CC_TIFF6         12  /* TIFF V6, multi-page   */
#define SIM_CC_TIFF6_PAGE    13  /* TIFF V6, single page  */
#define SIM_CC_TIFF_SINGLE_STRIP  14  /* raster in a single strip */
#define SIM_CC_MODCA_FORM    15  /* MODCA-P form overlay */

#define SIM_CC_IOCA_IRM      30  /* IRM version of IOCA, non-standard */
#define SIM_CC_TIFF_IRM      31  /* IRM version of TIFF, single-page  */
/* removed SIM_CC_RASTER 32 5/28/93 as unneeded */

#define SIM_CC_ASCII         40  /* flat ascii text              */
#define SIM_CC_EBCDIC        41  /* flat EBCDIC text             */
#define SIM_CC_BINARY        42  /* unformatted binary data      */
#define SIM_CC_TEXT          43  /* text, where code page, etc. is*/
                                 /* ..determined by auxiliary    */
                                 /* ..information stored elsewhere*/
#define SIM_CC_PPDS          44  /* printer data stream (text)   */

/* fax formats */

#define SIM_CC_TIFF_G3_STANDARD  50  /* standard fax, TIFF header */
#define SIM_CC_TIFF_G3_FINE  51  /* higher resolution fax        */
                                 /* standard: approx 100 dpi     */
                                 /* fine: approx 200 dpi         */
/* * * MORE FAX CATEGORIES MAY BE DEFINED * * */

/* bitmaps */

#define SIM_CC_OS2V2_BMP     70  /* OS/2 2.0 bitmap              */
#define SIM_CC_OS2V13_BMP    71  /* OS/2 1.3 bitmap              */
#define SIM_CC_OS2V12_BMP    72  /* OS/2 1.2 bitmap              */
#define SIM_CC_WINV3_BMP     73  /* MS Windows V3 bitmap         */

/* command files/programs */

#define SIM_CC_OS2EXE        90  /* executable program (OS/2)    */
#define SIM_CC_OS2CMD        91  /* command file (OS/2)          */
#define SIM_CC_OS2DLL        92  /* DLL (OS/2)                   */
#define SIM_CC_AIXEXE        93  /* executable program (AIX)     */
#define SIM_CC_AIXCMD        94  /* command file (AIX)           */

/* spreadsheets */

#define SIM_CC_WKS          120  /* standard spreadsheet format  */
#define SIM_CC_WG1          121  /* graphics, from 1-2-3/G       */
#define SIM_CC_EXCEL        122  /* Microsoft Excel              */

/* word processing formats */

#define SIM_CC_WP           150  /* WordPerfect format           */
#define SIM_CC_WORD         151  /* Microsoft Word format        */
#define SIM_CC_PRS          152  /* Freelance presentation       */
#define SIM_CC_DESCRIBE     153  /* Describe text editor         */

/* image formats */

#define SIM_CC_PCX          200  /*                              */
#define SIM_CC_RFT          201  /* IBM RFT&GML.DCA              */
#define SIM_CC_TARGA        202  /*                              */

/* others */
#define SIM_CC_POSTSCRIPT   300  /* postscript data              */
#define SIM_CC_BKMGR_READ   301  /* BookManager read format      */

/* ImagePlus application-specific data types */

#define SIM_CC_FRN_NOTE     400  /* application note log         */
#define SIM_CC_FRN_HISTORY  401  /* application history log      */

#define SIM_CC_USER  4096  /* start custom user/appl formats here */
```

Figure 4. VisualInfo Document Types (from FRNPCAPI.H)

SimLibCatalogObject

A VisualInfo Folder Manager API

This API is used to store data from an external file or, in other words, to import data.

The *SimLibCatalogObject* API is a versatile function for storing objects using various combinations of parameters. SimLibCatalogObject can:

- Create or reuse item parts
- Create or reuse items
- Set the content class of an item part
- Leave the object in opened or closed status

See *VisualInfo Application Programming Reference, Volume 1* for a detailed description of SimLibCatalogObject and SimLibCreateItem.

SimLibCatalogObject does not check if the data to be stored actually has the specified content class, so you should do your own checking before storing. You can use the Image Services Working Set API *SimWsDetermineCC* to determine its content class by analyzing the bit pattern it contains. See *VisualInfo Application Programming Reference, Volume 2* for a description of SimWsDetermineCC.

4.4 API Type Overview

Refer to *VisualInfo Application Programming Reference, Volume 1* for a detailed description of the Folder Manager APIs. Refer to *VisualInfo Application Programming Reference, Volume 2* for a detailed description of Image Services APIs.

See Appendix C, “IP3IMPRT Source Code” on page 97 for listings of IP3IMPRT.C, and the files IP3IMPRT.H, IP3IMPRT.MAK and IP3IMPRT.DEF.

4.5 Implementation Hints and Tips

The IMPORT process itself is a non-interactive process. There is no need for OS/2 Presentation Manager functions to be invoked. For that reason, the program IP3IMPRT has *WINDOWCOMPAT* specified in its .DEF file.

Chapter 5. The Display Function

This chapter describes a sample program that illustrates some of the VisualInfo facilities that you might use to display a document from the VisualInfo library. It includes Design Objectives and a description of the Program Logic with references to the API calls used. Points of interest raised during the development of the sample are also covered.

5.1 Design Objectives

One of the main functions of an image system is to display documents and objects of any type. Although this sample has been designed to display MO:DCA documents (the most common image format used in IBM image systems) it can readily be modified to display other format types.

The IP3DISP function was designed with the following in mind:

- To display an existing document stored on a VisualInfo object server.
- To be used as a building block to assist with the development of a larger application. You can change or enhance it as you wish.
- The program is a stand-alone version, which means that it runs independently and performs its own logon and logoff and so on. Normally in an application you would make logon and logoff an external one-time function to improve the performance of this module.
- The program is executed from the command line. IP3DISP is an OS/2 executable file with a single input parameter, the item ID of the document to be displayed.

5.2 Program Logic

The source code for the display function program is called IP3DISP.C. Please see Appendix D, “IP3DISP Source Code” on page 113 for a listing of this program.

The IP3DISP function is executed from the command line. The only input parameter is the document identification (DocID). It is always 16 bytes long. Using the Query Manager of DB2/2 you can find the document identification of an existing document. The table name may not be the same in every VisualInfo installation. It is one of the AVT0000x tables in the library server database. In our system the table of ItemIDs for NOINDEX Index Class was AVT00006. Another way to get a list of DocIDs is to run the the sample program IP3SRCHD.EXE. See Chapter 2, “The Search Function” on page 3 for details.

IP3DISP execution example:

```
IP3DISP /D=W8627539A9576113
```

Return codes and messages are sent to a message box window.

There are some special Display prototypes (FIWSDSP.H, delivered with VisualInfo), which we include as a header file.

You can refer to parts related to this sample code in Appendix D, “IP3DISP Source Code” on page 113.

The structure of the main program and flow is as follows:

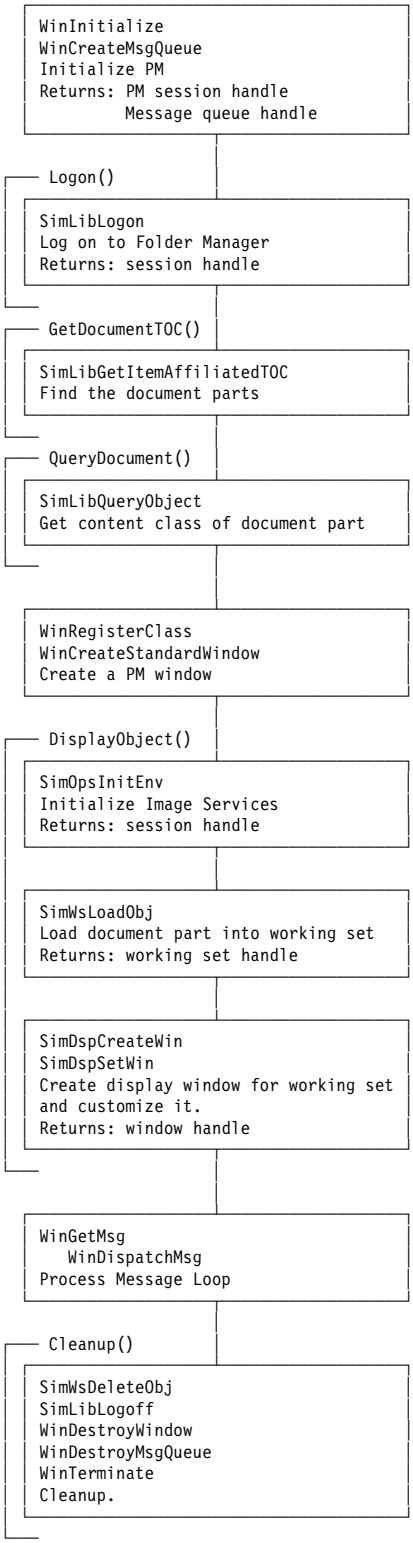


Figure 5. Program Flow for IP3DISP

WinInitialize

An OS/2 PM API

This function is called to obtain an anchor block handle and initialize Presentation Manager by calling WinInitialize.

The Display Services APIs are a part of the Image Services APIs of VisualInfo. As mentioned in *VisualInfo Application Programming Guide, Volume 2*, the Image Services APIs may only be invoked from Presentation Manager applications.

WinCreateMsgQueue

An OS/2 PM API

A Presentation Manager message queue is created to handle mouse and keyboard input.

WinRegisterClass and WinCreateStdWindow

OS/2 PM APIs

These APIs create a PM Window and specify its Window Procedure. We need to display a PM window to provide the user with a mechanism to terminate IP3DISP once the document has been viewed.

SimLibLogon

A VisualInfo Folder Manager API

The SimLibLogon API is used to establish a session with the Folder Manager. The initial values for this API are taken from the .H file (IP3DISP.H), where they are hard-coded. You could pass them to the main program via command line parameters if you wished with a small change to the program.

SimLibGetItemAffiliatedTOC

A VisualInfo Folder Manager API

A document (item) can contain several parts, that is, physical objects or binary large objects (BLOBs) stored on the object server. For example, a document (item), could consist of the relevant pages, an annotation, and so on, as parts. In order to find all item parts which make up a document, it is necessary to retrieve the table of contents for the document. The program saves the object structure of the first document part in the Object Handle Structure Obj. The object handle is then used in the calls to SimLibQueryObject and SimWsLoadObj.

SimLibQueryObject

A VisualInfo Folder Manager API

This function is called to determine the content class of the document to be displayed. The content class determines the format of a document (MO:DCA, TIFF and so on). We need this content class later for loading an object into a working set and displaying it correctly. The SimLibQueryObject Folder Manager API uses the object structure returned from SimLibGetItemAffiliatedTOC above, and stores its own return value in the variable ulObjConCls.

SimOpsInitEnv

A VisualInfo Image Services API

Before using Image Services within a program it is necessary to initialize the Image Services environment. In this sample program, the parameters used for the SimOpsInitEnv API are coded in the related header file (IP3DISP.H).

Please see “SimOpsInitEnv” on page 14 for additional information on this API call.

Since the document will be loaded from the server, the pszOManager parameter pointing to the Object Manager component of the Folder Manager must be provided (FRNOFO).

SimWsLoadObj

A VisualInfo Image Services API

The purpose of this function is to load the document into a working set. The SimWsLoadObj API creates a working set, retrieves the object from a source, in our case from the object server, and stores it in the working set. In order to read the data from the server we must specify SIMSRC_SERVER_OBJ for the parameter srcData. A structure of the type SIMSRCSERVEROBJ will be referenced to access to the server objects. This structure contains the following fields, which should be initialized as follows prior to the SimWsLoadObj call:

```
pDataSrc.hSession   = hSession;
pDataSrc.Obj        = Obj;
pDataSrc.ulAccess   = SIM_ACCESS_SHARED_READ;
pDataSrc.ulPriority = OM_NORMAL_PRIORITY;
```

You must define FIWS_SERVER in the program prior to the include for FIWSWS.H.

SimDspCreateWin

A VisualInfo Image Services API

The API SimDspCreateWin is used to create a new Image Services display window in order to view the working set data that contains the document.

SimWsDeleteObj

A VisualInfo Image Services API

Used in the Cleanup Module

When we have finished working with the document, we delete it in order to free the memory.

SimLibLogoff

A VisualInfo Folder Manager API

This function performs the logoff from the VisualInfo Folder Manager.

SimLibFree

A VisualInfo Folder Manager API

This function is called to free memory that was allocated by the Folder Manager SimLibLogon, SimLibGetItemAffiliatedTOC or a SimLibQueryObject API calls.

WinMessageBox

An OS/2 PM API

This function is called to display a message box window containing message text and a return code.

WinDestroyMsgQueue

An OS/2 PM API

The Presentation Manager message queue will be destroyed.

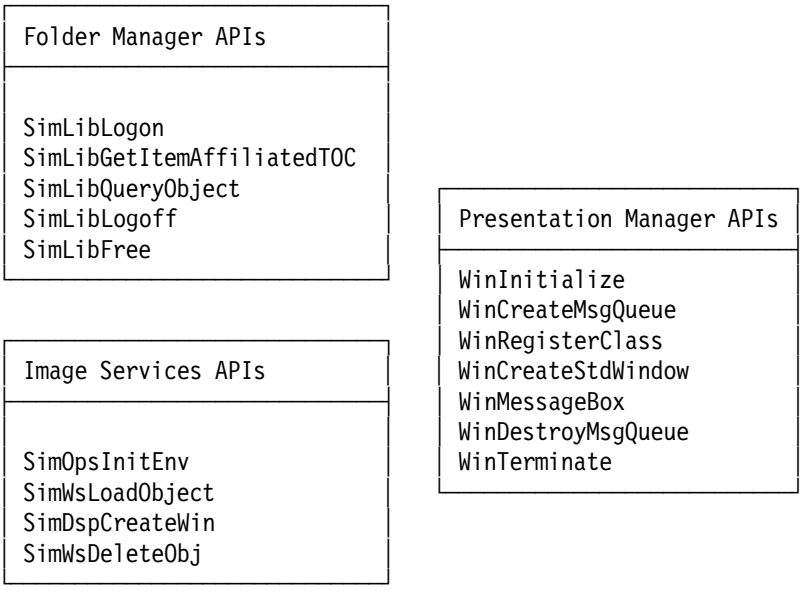
WinTerminate

An OS/2 PM API

This API terminates the use of Presentation Manager.

5.3 API Summary

The following figure summarizes the API calls used and their category.



Detailed programming information is available in:

- *VisualInfo Application Programming Reference, Volume 1* (Common Interfaces)
- *VisualInfo Application Programming Reference, Volume 2* (Image Services Interfaces)
- *VisualInfo Application Programming Guide, Volume 1*
- *VisualInfo Application Programming Guide, Volume 2*

5.4 Implementation Hints and Tips

- When you load a working set it will automatically be created. Therefore there is no need to use the API SimWsCreateObj.
- IP3SCAN was written with the bare minimum of function required to make it a PM program with a message queue and window procedure. A more typical program found in a complete application would provide user access to facilities through an action bar and pull-down menus from its main window.

Chapter 6. The Print Function

This chapter describes the IP3PRINT function that we developed for the VisualInfo environment. It includes Design Objectives and a description of the Program Logic with references to the API calls used. Points of interest raised during the development are also covered in hints and tips.

6.1 Design Objectives

Print functions can be a mixed blessing in an electronic image system environment. If overused they can create a paper storage problem. The objective of an electronic image system is to reduce paperwork, and file it electronically. The printing of documents can create a paper storage problem if used excessively. Ideally the only time a hardcopy of an electronically stored document should be required is for external purposes such as a customer copy or to be used as evidence in a courtroom. However, these issues aside, we recognize the need for a print function and have provided the IP3PRINT sample for this purpose.

The IP3PRINT function was designed with the following considerations:

- The general objective is to print a document stored on a VisualInfo object server.
- This sample code is designed to be used as a building block to assist you in application development. You can change or enhance it.
- The program is a stand-alone version. It runs independently, performs its own logon and logoff and so on. You would normally make logon and logoff an external one-time function in an application to improve performance.
- The program is executed from the command line. The only input parameter is the document ID. This allows you to use this sample code for testing purposes of the VisualInfo APIs.
- The registration of the printer and the creation of the printer options profile (POP) must be done prior to executing the print function. We decided not to do the registration of the printer and creation of the POP within the IP3PRINT program, because this function only has to be executed once. Normally this is done once during the setup phase of VisualInfo and is performed by a system administrator. The registration and creation of the POP can be done by the Client Application, or by an API-program that you must create. Chapter 7, “Create a Printer Option Profile” on page 45 shows how to create the printer option profile for IP3PRINT.

6.2 Program Logic

The source code for the print function program is called IP3PRINT.C. Refer to Appendix E, “IP3PRINT Source Code” on page 137 for a listing of this program and associated components.

The IP3PRINT function is executed from the command line. The only input parameter is the document identification (DocID). The DocID is always 16 bytes long. Using the Query Manager of Database 2 OS/2 you can find the document identification of an existing document. Querying the AVT0000x tables in the library server database will give you some valid DocIDs. Try AVT00006 for a start. Another way of obtaining DocIDs is to run the sample program IP3SRCHD.EXE to do a search for documents. See Chapter 2, “The Search Function” on page 3 for more details.

IP3PRINT execution example:

```
IP3PRINT W8627539A9576113
```

Return codes and Messages are displayed in a PM message box.

There are some special print prototypes (FIWSPRT.H, delivered with VisualInfo), which we include in the header file.

You can refer to the source and components related to this sample code in Appendix E, “IP3PRINT Source Code” on page 137.

The structure of the main program is as follows:

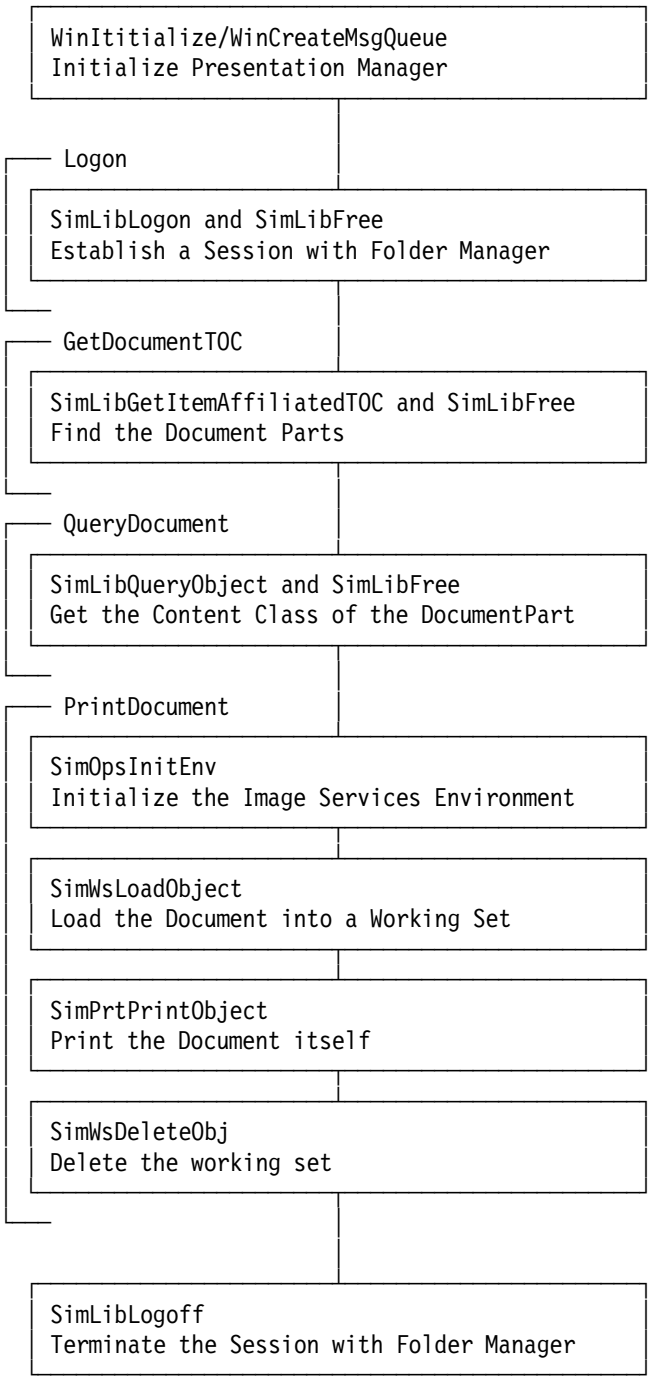


Figure 6. Program Flow for IP3PRINT

WinInitialize

An OS/2 PM API

This function is called to obtain an anchor block handle and initialize Presentation Manager.

The Print Services APIs are a part of the Image Services APIs of VisualInfo. As mentioned in *VisualInfo Application Programming Guide, Volume 2*, the Image

Services APIs may be invoked only from Presentation Manager applications. In our example it is necessary to initialize only the Presentation Manager.

WinCreateMsgQueue

Provided by OS/2 Presentation Manager

This API creates an OS/2 Presentation Manager message queue. This is not strictly required for Image Services, but is required so we can display messages through a WinMessageBox function.

SimLibLogon

A VisualInfo Folder Manager API

The SimLibLogon API is used to establish a session with the Folder Manager. The initial values for this API are taken from the .H file (IP3PRINT.H), where they are hard-coded. You could change the program to pass parameters to the main program via the command line if you wanted that flexibility.

SimLibGetItemAffiliatedTOC

A VisualInfo Folder Manager API

A document (item) can contain several parts, that is physical objects or binary large objects (BLOBs) which can represent single pages or multiple pages. These are called item parts and are stored on the object server and comprise the document (which may consist of one or more parts) and optionally an annotation part and so on. In order to find all the item parts which belong to an existing document, it is necessary to retrieve the table of contents for a specific document. The program saves a pointer to the object structure of the first document part in the variable Obj. In the SimLibQueryObject and SimWsLoadObj APIs this returned pointer to the object structure is used to identify the document part that is to be printed.

SimLibQueryObject

A VisualInfo Folder Manager API

This function is called to determine the content class of the document to be printed. The content class determines the format of a document (for example, MO:DCA, TIFF and so on).

We need the content class later for loading an object into a working set. The SimLibQueryObject Folder Manager API uses the Obj pointer returned from SimLibGetItemAffiliatedTOC, and stores the content class return value in the variable ulObjConCls.

SimOpsInitEnv

A VisualInfo Image Services API

The first time you use Image Services within a program it is necessary to initialize the environment to allow the following APIs to function. Whenever you use Image Services APIs you have to initialize the environment as a first step. In this sample code, the parameters used for the SimOpsInitEnv API are coded in the related header file (IP3PRINT.H).

In parameter 6 (pszOManager) we specify the name of the DLL containing the Object Manager component of the Folder Manager (FRNOFO).

SimWsLoadObj

A VisualInfo Image Services API

The purpose of this function is to load the document into the working set. The SimWsLoadObj API creates a working set, retrieves the object from a source, in our case from the object server, and stores it into the working set created. In order to read the data from the server we must specify SIMSRC_SERVER_OBJ for the parameter srcData. A structure of the type SIMSRCSERVEROBJ will be referenced to access to the server objects. This structure is initialized as follows in our sample:

```
pDataSrc.hSession   = hSession;
pDataSrc.Obj        = Obj;
pDataSrc.ulAccess   = SIM_ACCESS_SHARED_READ;
pDataSrc.ulPriority = OM_NORMAL_PRIORITY;
```

You must define FIWS_SERVER in the program prior to the include for FIWSWS.H.

SimPrtPrintObject

A VisualInfo Image Services API

The above API performs the actual printing of the document (item). The first parameter (pszPOPDesc) is the description of the printer profile (POP), which is used by Print Services to define the manner in which the document should be printed. In this sample code we use the pszPOPDesc created using the IP3POP sample program described in Chapter 7, “Create a Printer Option Profile” on page 45. You may wish to define the printer option profile (POP) as a variable, if you enhance or rewrite this program.

SimWsDeleteObj

A VisualInfo Image Services API

When we have finished with the working set we delete it in order to free the memory.

SimLibLogoff

A VisualInfo Folder Manager API

This function performs the logoff from the VisualInfo Folder Manager.

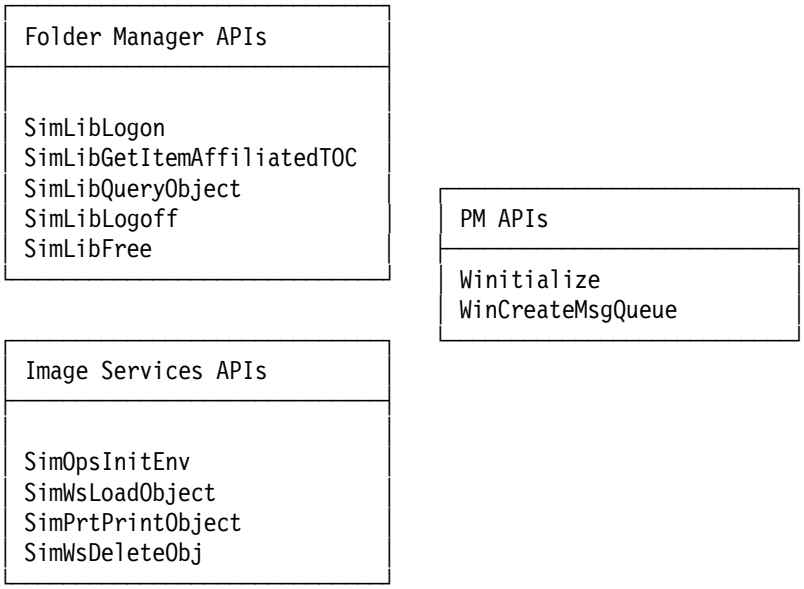
SimLibFree

A VisualInfo Folder Manager API

This function is called to free memory that was allocated by the Folder Manager SimLibLogon, SimLibGetItemAffiliatedTOC or SimLibQueryObject API calls.

6.3 API Summary

The following figure gives you an overview of the APIs and their categories used in this sample code:



For further information refer to:

- *VisualInfo Application Programming Reference, Volume 1* (Common Interfaces)
- *VisualInfo Application Programming Reference, Volume 2* (Image Services Interfaces)
- *VisualInfo Application Programming Guide, Volume 1*
- *VisualInfo Application Programming Guide, Volume 2*

6.4 Implementation Hints and Tips

- Image Services APIs can be invoked only from Presentation Manager programs. If you don't initialize Presentation Manager, the document will not print.
- When you load a working set it will automatically be created. Therefore there is no need to use the API SimWsCreateObj.

Chapter 7. Create a Printer Option Profile

This chapter describes how to create a printer option profile (POP) for the VisualInfo environment. The printer option profile must be created before you can use the print function.

7.1 Design Objectives

The IP3POP function was designed to provide the following:

- Run stand-alone
- Query the existing printer option profiles
- Create a printer option profile

The program is executed from the command line and no parameters are required. The parameters supplied to create the POP are hard-coded in the program and match those required by the IP3PRINT program.

When IP3POP is executed, Image Services brings up a notebook window asking you to select a printer on your PS/2 system. If you try to run the program twice you will get a message from Image Services telling you that the POP already exists. If you wish to run the program again go into the VisualInfo Client Application and delete the POP called SAMPLEAPPRT.

7.2 Program Logic

The source code can be found in Appendix F, “IP3POP Source Code” on page 157. The IP3POP function is executed from the command line and does not require any parameters.

Execution example:

```
IP3POP
```

Return codes and messages are written to the screen in a dialog box.

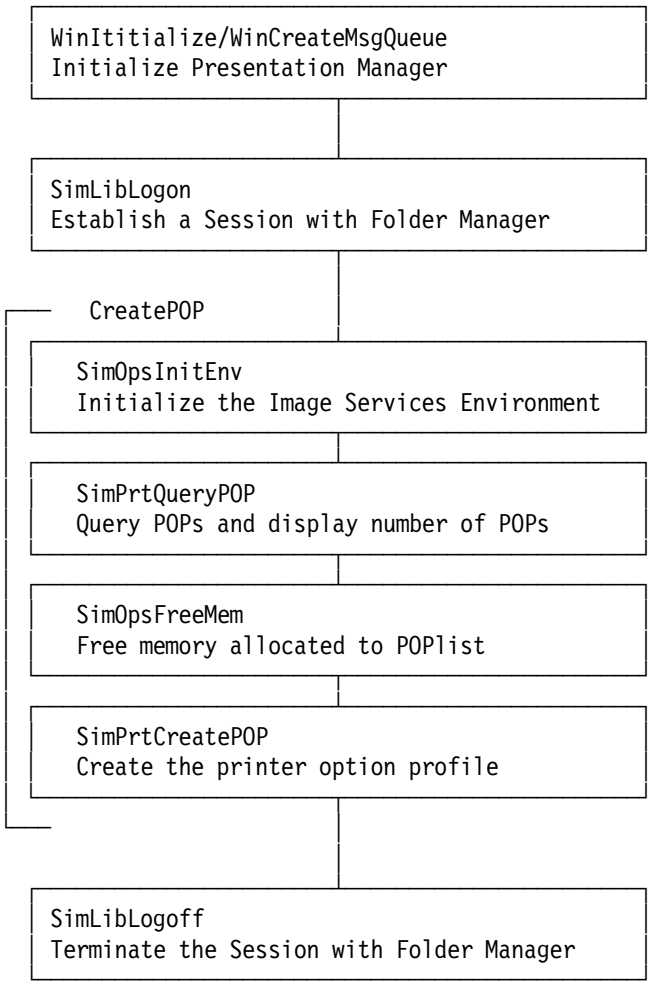


Figure 7. Program Flow for IP3POP

SimLibLogon

A VisualInfo Folder Manager API

The SimLibLogon API is used to establish a session with the Folder Manager. The initial values for this API are taken from the .H file (IP3PRINT.H), where they are hard-coded. It would be just as easy to pass them to the main program via command line parameters although for the purpose of this example a hard-coded POP name (SAMPLEAPPRT) is all that is required.

SimOpsInitEnv

A VisualInfo Image Services API

The first time you use Image Services within a program it is necessary to initialize it and to describe the environment. This is done with the SimOpsInitEnv API. The created environment is referred to as a *services instance*. Whenever you use Image Services APIs you have to initialize the environment as a first step. In this sample code, the parameters used for the SimOpsInitEnv API are coded in the header file (IP3POP.H).

SimPrtQueryPOP

A VisualInfo Image Services API

The purpose of this function is query the number of POPs defined to the system. The SimPrtQueryPOP returns the number of POPs and a pointer to an array of printer option profile elements. This sample does not do any processing on the POP elements, and simply displays the number of POPs for the user's information. The client application must free the memory containing the POPlist when finished with the information by issuing a SimOpsFreeMem API. This is performed in the IP3POP sample immediately after the SimPrtQueryPOP API.

SimPrtCreatePOP

A VisualInfo Image Services API

The SimPrtCreatePOP API creates a new printer option profile using a hard-coded value of "SAMPLEAPPRT" for parameter pszPOPDesc. The program then displays the return code for this operation and returns to the main program to log off.

SimLibLogoff

A VisualInfo Folder Manager API

This function performs the logoff from the VisualInfo Folder Manager.

7.3 Implementation Hints and Tips

- We originally intended to process the POP elements returned from the SimPrtQueryPOP API, and based upon that information make a decision on whether to create a new POP or not. However we decided not to do it in the interests of keeping it simple, and because the interactive window that Image Services invokes gives the user enough flexibility to create and change the information, and to cancel out if required.
- You should run IP3POP before running IP3PRINT.
- You can change and delete the printer option profiles created with IP3POP through the client application.

Chapter 8. The Export Function

This chapter describes a sample program that illustrates some of the VisualInfo facilities that you might use for exporting an object from a VisualInfo library to a workstation file.

8.1 Design Objectives

The export function is useful for loading and unloading a small number of objects for testing and demonstration purposes.

For loading and unloading a large number of objects, such as in a migration from one system to another, it would be better to use the interchange APIs. The VisualInfo interchange APIs have been designed for migrations and movement of large numbers of objects (and their related indexing information) from one system to another. See the Ip2Import and Ip2Export Folder Manager APIs for more information on this subject.

The IP3EXPRT sample was designed with the following considerations:

- To retrieve an existing document from the VisualInfo library and copy it to an OS/2 disk file.
- The program is a stand-alone version. That means it runs as an independent entity, performs its own logon and logoff and so on. In an application you would make logon and logoff an external one-time function to improve performance.
- The program is executed from the command line. IP3EXPRT is a .EXE file. The input parameters are the document identification and the name of the file, which is created by the program.

8.2 Program Logic

The source code for the export function is program is called IP3EXPRT.C. Please see Appendix G, “IP3EXPRT Source Code” on page 171 for a listing of this program and related components.

The IP3EXPRT function is executed from the command line. The following input parameters are required:

- /D=DocID

  This is always 16 bytes long. Using the Query Manager of DB2/2 you can find the Document Identification of existing documents. DocIDs/ItemIDs will be in one of the AVT0000x tables in the library server database. The table of ItemIDs for NOINDEX Index Class will always be AVT00006. Another way to get a list of DocIDs is to run the the sample program IP3SRCHD.EXE. See Chapter 2, “The Search Function” on page 3 for details.

- /F=Filename

  The second input parameter is the name of the file to be created. In this example TEST.MDA will be placed in the current directory.

Execution example:

```
IP3EXPRT /D=W8627539A9576113 /F=TEST.MDA
```

Return codes and messages are written to the screen.

The structure of the IP3EXPRT program is as follows:

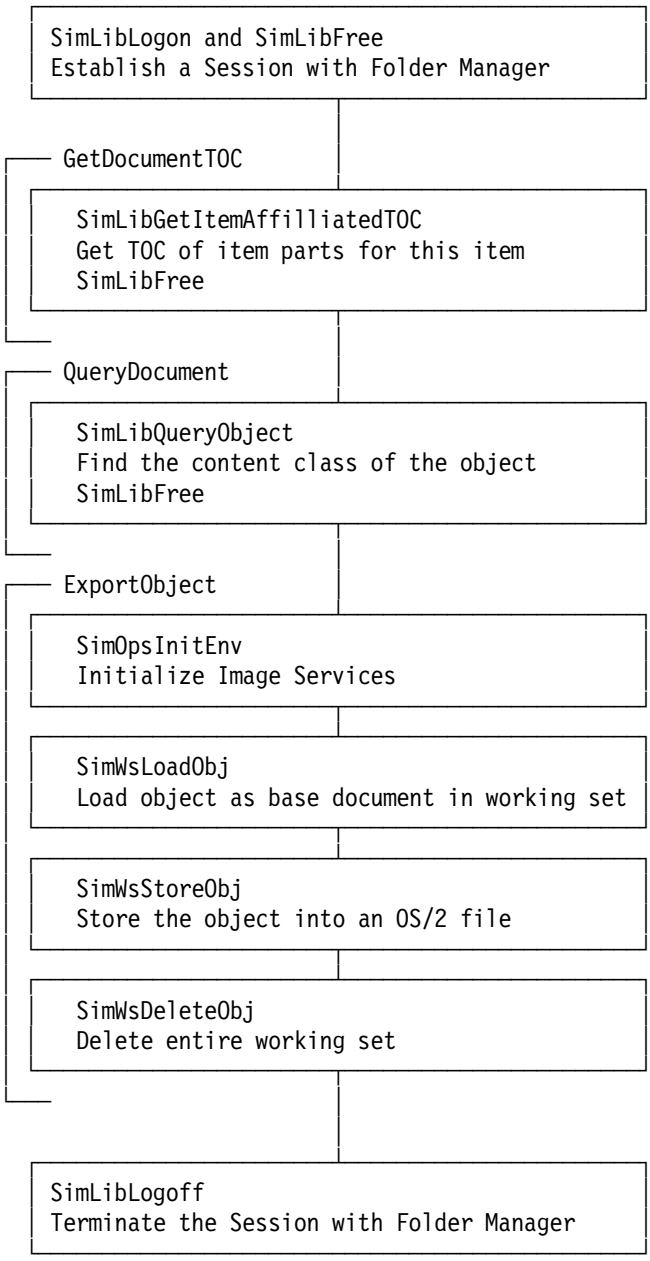


Figure 8. Program Flow for IP3EXPRT

SimLibLogon

A VisualInfo Folder Manager API

The SimLibLogon API is used to establish a session with the Folder Manager. The initial values for this API are taken from the .H file (IP3EXPRT.H), where they are hard-coded.

SimLibGetItemAffiliatedTOC

A VisualInfo Folder Manager API

A document (item) can contain several parts, that is, physical objects or binary large objects (BLOBs) stored on the object server. For example, a document (item), could consist of the relevant pages, an annotation, and so on, as parts. In order to find all item parts which make up a document, it is necessary to retrieve the table of contents for the document. The program saves the object structure of the first document part in the Object Handle structure Obj. The object handle is then used in the calls to SimLibQueryObject and SimWsLoadObj.

SimLibQueryObject

A VisualInfo Folder Manager API

This function is called to determine the content class of the document to be exported. The content class specifies the format of the document (MO:DCA, TIFF and so on). We need this content class later for loading an object into a working set. The API SimLibQueryObject API uses the Objectstructure returned from SimLibGetItemAffiliatedTOC. After executing the SimLibQueryObject API the object content class is stored in the variable ulObjConCls.

SimOpsInitEnv

A VisualInfo Image Services API

The first time you use Image Services within a program it is necessary to perform SimOpsInitEnv.

SimWsLoadObj

A VisualInfo Image Services API

The purpose of this function is to load the document into a working set. The SimWsLoadObj API creates a working set, retrieves the object from a source, in our case from the object server, and stores it in the working set. In order to read the data from the server we must specify SIMSRC_SERVER_OBJ for the parameter srcData. A structure of the type SIMSRCSERVEROBJ will be referenced to access to the server objects. This structure contains the following fields, which should be initialized as follows prior to the SimWsLoadObj call:

```
pDataSrc.hSession   = hSession;
pDataSrc.Obj        = Obj;
pDataSrc.ulAccess   = SIM_ACCESS_SHARED_READ;
pDataSrc.ulPriority = OM_NORMAL_PRIORITY;
```

You must define FIWS_SERVER in the program prior to the include for FIWSWS.H.

SimWsStoreObj

A VisualInfo Image Services API

This function calls the SimWsStoreObj API to write the object into a file. The way to do that is to use the value SIMSRC_FILENAME for the srcData parameter. The handle to the working set, the object content class is passed to the API as well as a pointer to the filename to be created. The pDataSrc parameter is the destination to store the data. In this case it is a file name and must be a pointer to a pointer as documented in the program for this function to work.

At this point if all is well the object will be written to the OS/2 file as specified in the /F parameter.

SimWsDeleteObj

A VisualInfo Image Services API

When we have finished with the working set we delete the entire object using the parameter SIMPURGE_ALL.

SimLibLogoff

A VisualInfo Folder Manager API

This function performs the logoff from the VisualInfo Folder Manager.

SimLibFree

A VisualInfo Folder Manager API

This function is called to free memory that was allocated by the SimLibLogon.

8.3 API Summary

The following figure summarizes the APIs used in this program:

Detailed Information is available in the following documents:

- *VisualInfo Application Programming Reference, Volume 1* (Common Interfaces)
- *VisualInfo Application Programming Reference, Volume 2* (Image Services Interfaces)
- *VisualInfo Application Programming Guide, Volume 1*
- *VisualInfo Application Programming Guide, Volume 2*

8.4 Implementation Hints and Tips

- When you load a working set it will automatically be created. You do not need to use the API SimWsCreateObj.
- The destination parameter pDataSrc must be a *pointer to a pointer* to a string which contains the filename. If you do not add this extra level of pointer indirection the API call will fail.

Appendix A. IP3SRCHD Source Code

A.1 Function index

DosSearchPath	
called by	**main** (IP3SRCHD.C, page 61)

exit	
called by	**main** (IP3SRCHD.C, page 61)

fclose	
called by	**main** (IP3SRCHD.C, page 61)

fopen	
called by	**main** (IP3SRCHD.C, page 61)

fprintf	
called by	**main** (IP3SRCHD.C, page 61)

Ip2ListAttrs	
called by	**main** (IP3SRCHD.C, page 61)

main (IP3SRCHD.C, page 61)	
calls	**DosSearchPath** **exit** **fclose** **fopen** **fprintf** **Ip2ListAttrs** **printf** **SimLibGetClassInfo** **SimLibListClasses** **SimLibListClassViews** **SimLibLogoff** **SimLibLogon** **SimLibSearch** **sprintf** **strcat** **strcpy** **stricmp** **strncpy** **strnicmp** **strpbrk**

printf	
called by	**main** (IP3SRCHD.C, page 61)

SimLibGetClassInfo	
called by	**main** (IP3SRCHD.C, page 61)

SimLibListClasses	
called by	**main** (IP3SRCHD.C, page 61)

SimLibListClassViews	
called by	**main** (IP3SRCHD.C, page 61)

SimLibLogoff	
called by	**main** (IP3SRCHD.C, page 61)

SimLibLogon	
called by	**main** (IP3SRCHD.C, page 61)

SimLibSearch	
called by	**main** (IP3SRCHD.C, page 61)

sprintf	
called by	**main** (IP3SRCHD.C, page 61)

strcat	
called by	**main** (IP3SRCHD.C, page 61)

strcpy	
called by	**main** (IP3SRCHD.C, page 61)

stricmp	
called by	**main** (IP3SRCHD.C, page 61)

strnicmp	
called by	**main** (IP3SRCHD.C, page 61)

strncpy	
called by	**main** (IP3SRCHD.C, page 61)

strpbrk	
called by	**main** (IP3SRCHD.C, page 61)

A.2 IP3SRCHD.H (06/29/94 14:32:36)

```
1   /*****************************************************************************/
2   /*                                                                           */
3   /* MODULE: IP3SRCHD.H                                                        */
4   /*                                                                           */
5   /* Description  : Header file for module IP3SRCHD.C.                        */
6   /*                                                                           */
7   /*****************************************************************************/
8
9   #define INCL_DOS
10
11  #include <os2.h>
12  #include <bsememf.h>
13  #include <string.h>
14  #include <stdio.h>
15  #include <stdlib.h>
16  #include <frnptype.h>
17  #include <frnpcapi.h>
18  #include <frnpfi.h>
19  #include <frnpfi2.h>
20
21  #define IP3INFO "IP3INFO.TXT"
22  #define IP3RESU "IP3RESU.TXT"
23
24  /*****************************************************************************/
25  /* End of IP3SRCHD.H                                                         */
26  /*****************************************************************************/
```

A.3 IP3SRCHD.C (06/23/94 15:24:26)

main (IP3SRCHD.C, page 61)	
calls	**DosSearchPath** **exit** **fclose** **fopen** **fprintf** **Ip2ListAttrs** **printf** **SimLibGetClassInfo** **SimLibListClasses** **SimLibListClassViews** **SimLibLogoff** **SimLibLogon** **SimLibSearch** **sprintf** **strcat** **strcpy** **stricmp** **strncpy** **strnicmp** **strpbrk**

```
 1   #pragma langlvl(extended)
 2
 3   /*****************************************************************************/
 4   /* DISCLAIMER OF WARRANTIES:                                                 */
 5   /* -------------------------                                                 */
 6   /* The following [enclosed] code is sample code created by IBM               */
 7   /* Corporation.  This sample code is not part of any standard IBM product    */
 8   /* and is provided to you solely for the purpose of assisting you in the     */
 9   /* development of your applications.  The code is provided "AS IS",          */
10   /* without warranty of any kind.  IBM shall not be liable for any damages    */
11   /* arising out of your use of the sample code, even if they have been        */
12   /* advised of the possibility of such damages.                               */
13   /*****************************************************************************/
14
15    /*********************************************************************/
16    /***                                                               ***/
17    /*** IP3SRCHD.C                                                    ***/
18    /***                                                               ***/
19    /*** Search for document by arguments                              ***/
20    /***                                                               ***/
21    /***    Arguments;                                                 ***/
22    /***                                                               ***/
23    /***            /I=IndexClassName     Index Class Name             ***/
24    /***            /AAttributeName=value Attribute Value Selection    ***/
25    /***            /F=OutputFileName     Output File Name and Path    ***/
26    /***                                                               ***/
27    /***    Returncodes:                                               ***/
28    /***                                                               ***/
29    /***                    0     everything fine                      ***/
30    /***                    4     no documents selected                ***/
31    /***                    8     Error for file IP3INFO.TXT           ***/
32    /***                   12     Invalid or missing parameter         ***/
33    /***                 xxxx     Severe error  in     SB API (API rc) ***/
34    /***                                                               ***/
35    /***               * = multiple occurence                          ***/
36    /***                                                               ***/
37    /*********************************************************************/
38    /*** created:                                 W. Stanzl  930915    ***/
39    /***                                                               ***/
40    /***                                                               ***/
41    /*********************************************************************/
42
43    #include "ip3srchd.h"
```

```
44  /*********************************************************************/
45  /*** Ip3Srchd.C                                                    ***/
46  /*********************************************************************/
47
48  int main(int argc, char * argv[]) {
49    HSESSION              hSession = 0;
50    int                   cntarg;
51    int                   iSelCrit;
52    int                   iulParam2;
53    int                   iCntParmF;
54    int                   iCntParmI;
55    int                   iCntParmA;
56    int                   iCntParmError;
57    int                   iCntSearchArgs;
58    int                   iItemId;
59    int                   iAttr;
60    int                   iOp;
61    int                   iAttrLen;
62    int                   iClassViews;
63    char                  *pOpPos;
64    char                  pszArgKey[3] = "";
65    char                  pszTransOp[4][4] = {"=EQ", "<LT", ">GT", "!NE"};
66    char                  pszSS[1024];
67    char                  pszS1[1024];
68    ULONG                 ulParam2ListClasses;
69    ULONG                 ulCntAttrs;
70    FILE                  *hIp3ResultTxt;
71    char                  *pszNamIp3Result;
72    char                  pszNamIp3ResultTxt[256];
73    char                  *pszNamIndexClass;
74    char                  * pszSelCrit[16];
75
76    APIRET                DosSearchPathRC;
77
78    BITS                          fClassOptions;
79    PASYNCCTLSTRUCT               pAsyncCtl;
80    PRCSTRUCT                     pRC;
81    RCSTRUCT                      RCStruct;
82    PNAMESTRUCT                   pNameStruct;
83    PNAMESTRUCT                   pAttrNames;
84    PNAMESTRUCT                   pClassViews;
85    char                          pszLanguageCode[4] = "ENU";
86    USHORT                        usClassType;
87    PCLASSINFOSTRUCT              pClassInfoStruct;
88
89    LIBSEARCHCRITERIASTRUCT       pCriteria[16];
90    int                           iCntItemId;
91    PITEMID                       pItemId;
92    USHORT                        usNumCriteria;
93
94  /*********************************************************************/
95  /*** Get Parameter values                                          ***/
96  /*********************************************************************/
97  /*** Build parameter table                                         ***/
98  /***                                                             ***/
99  /*********************************************************************/
100 /***      /F=OutFileName      into pszNamIp3Result                 ***/
101 /***      /I=IndexClassName   into pszNamIndexClass                ***/
102 /***      /Aattribute=value   into pszSelCrit[iSelCrit]            ***/
103 /***                                                             ***/
104 /*********************************************************************/
105
106   printf("%s started.\n", argv[0]);
107
108   printf("Getting parameter values\n");
109
```

```
110     iCntParmError = 0;
111     iSelCrit = 0;
112     iCntParmF = 0;
113     iCntParmA = 0;
114     iCntParmI = 0;
115
116     for (cntarg = 1; cntarg < argc; cntarg++) {
117        printf(" Parameter %i: %s \n", cntarg, argv[cntarg]);
118
119        strncpy(pszArgKey,argv[cntarg],2);
120
121        if (stricmp(pszArgKey,"/F") == 0) {
122           pszNamIp3Result = argv[cntarg]+3;
123           iCntParmF++;
124           printf(" /F: output file name is %s \n",pszNamIp3Result);
125        }
126
127        else if (stricmp(pszArgKey,"/I") == 0) {
128           iCntParmI++;
129           pszNamIndexClass = argv[cntarg]+3;
130           printf(" /I: index class name is %s \n",pszNamIndexClass);
131        }
132
133        else if (stricmp(pszArgKey,"/A") == 0) {
134           iCntParmA++;
135           iSelCrit++;
136           pszSelCrit[iSelCrit] = argv[cntarg]+2;
137           printf(" /A: selection value is %s \n",pszSelCrit
138                 [iSelCrit]);
139        }
140
141        else {
142           printf(" Invalid parameter : %s \n",argv[cntarg]);
143           iCntParmError++;
144        }
145     }
146
147     if (iCntParmI == 0) {
148        printf(" Parameter /I (Index Class) missing\n");
149        iCntParmError++;
150     }
151
152     if (iCntParmA == 0) {
153        printf(" Parameter /A (Selection Arguments) missing\n");
154        iCntParmError++;
155     }
156
157     if (iCntParmF == 0) {
158        printf(" Parameter /F (Output File Name) missing\n");
159        iCntParmError++;
160        }
161
162     if (iCntParmI > 1) {
163        printf(" Parameter /I (Index Class) specified more than once\n");
164        iCntParmError++;
165     }
166
167     if (iCntParmError > 0) {
168        printf(" Parameter error.");
169        exit(12);
170     }
171
172  /*********************************************************************/
```

```
173  /*** Logon to Library Server                                          ***/
174  /***                                                                  ***/
175  /*** Api: SimLibLogon                                                 ***/
176  /**********************************************************************/
177
178    printf("Performing local logon\n");
179
180    pRC = &RCStruct;
181    pAsyncCtl = NULL;
182
183    SimLibLogon("LIBSRVR2",
184                "SAMPLE",
185                "FRNADMIN",
186                "PASSWORD",
187                NULL,
188                NULL,
189                NULL,
190                SIM_SS_NORMAL,
191                pAsyncCtl,
192                pRC);
193
194    printf(" SimLibLogon complete.\n");
195
196    if (pRC->ulRC == SIM_RC_OK) {
197       printf(" ulRC from SimLibLogon is SIM_RC_OK.\n");
198    }
199    else {
200       printf(" ulRC from SimLibLogon is %i\n", pRC->ulRC);
201    }
202
203    if (pRC->usParam == 0) {
204       hSession = (HSESSION)pRC->ulParam1;
205       printf(" hSession has been assigned value %X\n", hSession);
206    }
207
208  /**********************************************************************/
209  /*** List all index classes to get all Index Class Ids              ***/
210  /***                                                                ***/
211  /*** Note:                                                          ***/
212  /***    Namestruct.szName is not the real name of the index class   ***/
213  /***    but just the name of it known by FaxRouter/2. So don't      ***/
214  /***    use this one but rather use SimLibGetClassInfo to get it.   ***/
215  /***                                                                ***/
216  /*** Api: SimLibListClasses                                         ***/
217  /**********************************************************************/
218
219    printf("Performing SimLibListClasses \n");
220
221    fClassOptions = SIM_ENUM_APPL;
222    pAsyncCtl = NULL;
223    pRC = &RCStruct;
224    printf(" hSession is %X\n", hSession);
225
226    SimLibListClasses(hSession, fClassOptions,
227                      pAsyncCtl, pRC);
228
229    printf(" SimLibListClasses complete.\n");
230
231    if (pRC->ulRC == SIM_RC_OK) {
232       printf(" ulRC from SimLibListClasses is SIM_RC_OK.\n");
233    }
234    else {
235       printf(" ulRC from SimLibListClasses is %i\n", pRC->ulRC);
236    }
```

```
237 |
238 |   printf(" ulParam1 from SimLibListClasses is %p\n", pRC -> ulParam1);
239 |   printf(" ulParam2 from SimLibListClasses is %p\n", pRC -> ulParam2);
240 |
241 |   if (pRC -> ulParam1 > 0 &&
242 |  ┌    pRC -> ulParam2 > 0) {
243 |  |
244 |  |    pNameStruct = (VOID *)(pRC->ulParam1);
245 |  |    ulParam2ListClasses = pRC->ulParam2;
246 |  └}
247 |
248 |   printf("SimLibListClasses ended.\n");
249 |
250 |  /*********************************************************************/
251 |  /*** Perform SimLibGetClassInfo to get the index class name        ***/
252 |  /*** select index class by name, keep index class id.              ***/
253 |  /***                                                              ***/
254 |  /*** Api: SimLibGetClassInfo                                      ***/
255 |  /*********************************************************************/
256 |
257 |   printf("Performing SimLibGetClassInfo\n");
258 |
259 |   usClassType = SIM_INDEXCLASSID;
260 |   pAsyncCtl = NULL;
261 |   pRC = &RCStruct;
262 |
263 |  ┌for (iulParam2 = 0; iulParam2 < ulParam2ListClasses; iulParam2++ ) {
264 |  |    printf ("%i %s\n", (pNameStruct+iulParam2)->usID,
265 |  |                        &((pNameStruct+iulParam2)->szName));
266 |  |    SimLibGetClassInfo(hSession,
267 |  |                       usClassType,
268 |  |                       (pNameStruct+iulParam2)->usID,
269 |  |                       pAsyncCtl,
270 |  |                       pRC);
271 |  |
272 |  |    printf(" usParam  from SimLibGetClassinfo is %i\n", pRC -> usParam );
273 |  |    printf(" ulParam1 from SimLibGetClassInfo is %i\n", pRC -> ulParam1);
274 |  |
275 |  |   ┌if (pRC->usParam == 1) {
276 |  |   |   pClassInfoStruct = (PCLASSINFOSTRUCT)pRC->ulParam1;
277 |  |   |   printf("               Class %i has name %s\n",
278 |  |   |          (pNameStruct+iulParam2)->usID,
279 |  |   |          &(pClassInfoStruct->szClassName));
280 |  |   |   if(! stricmp(pszNamIndexClass,
281 |  |   |  ┌           (PSZ)&(pClassInfoStruct->szClassName))) {
282 |  |   |  |    printf("              This is the Index Class you specified.\n");
283 |  |   |  └}
284 |  |   └}
285 |  └}
286 |
287 |  /*********************************************************************/
288 |  /*** Perform SimLibListClassViews to get all index class view names. */
289 |  /***                                                              ***/
290 |  /*** Api: SimLibListClassViews                                     ***/
291 |  /*********************************************************************/
292 |
293 |   printf("Performing SimLibGetClassViews\n");
294 |
295 |   pAsyncCtl = NULL;
296 |   pRC = &RCStruct;
297 |
298 |  ┌for (iulParam2 = 0; iulParam2 < ulParam2ListClasses; iulParam2++ ) {
299 |  |    printf (" Views of %i %s\n", pNameStruct->usID, &(pNameStruct->szName));
```

```
300     SimLibListClassViews(hSession,
301                          pNameStruct->usID,
302                          pAsyncCtl,
303                          pRC);
304     printf(" usParam  from SimLibListClassViews is %i\n", pRC -> usParam );
305     printf(" ulParam1 from SimLibListClassViews is %i\n", pRC -> ulParam1);
306     printf(" ulParam2 from SimLibListClassViews is %i\n", pRC -> ulParam2);
307     if (pRC->ulParam2 > 0) {
308        pClassViews = (PNAMESTRUCT)pRC->ulParam1;
309        for (iClassViews = 0; iClassViews < pRC->ulParam2; iClassViews++) {
310           printf(" Class %i has a View with ID %i and Name %s\n",
311                  pNameStruct->usID,
312                  (pClassViews+iClassViews)->usID,
313                  &((pClassViews+iClassViews)->szDescription));
314        }
315     }
316     pNameStruct++;
317   }
318
319  /*************************************************************************/
320  /*** List all index attributes to get attribute id.                    ***/
321  /***                                                                   ***/
322  /*** Result is structure type NAMESTRUCT, *pAttrNames                  ***/
323  /***                         number or occurences       ulCntAttrs   ***/
324  /***                                                                   ***/
325  /*** Api: Ip2ListAttrs                                                ***/
326  /*************************************************************************/
327
328    printf("Performing Ip2ListAttrs\n");
329
330    pAsyncCtl = NULL;
331    pRC = &RCStruct;
332
333    Ip2ListAttrs(hSession,
334                 pAsyncCtl,
335                 pRC);
336    printf(" Ip2ListAttrs complete.\n");
337
338    if (pRC->ulRC == SIM_RC_OK) {
339       printf(" ulRC from Ip2ListAttrs is SIM_RC_OK.\n");
340    }
341    else {
342       printf(" ulRC from Ip2ListAttrs is %i\n", pRC->ulRC);
343    }
344
345    printf(" ulParam1 from Ip2ListAttrs is %p\n", pRC -> ulParam1);
346    printf(" ulParam2 from Ip2ListAttrs is %p\n", pRC -> ulParam2);
347
348    if (pRC -> ulParam1 > 0 &&
349        pRC -> ulParam2 > 0) {
350       pAttrNames = (PNAMESTRUCT)(pRC->ulParam1);
351       ulCntAttrs = pRC->ulParam2;
352    }
353
354    for (iAttr = 0; iAttr < ulCntAttrs; iAttr++ ) {
355       printf ("%i %s\n",
356              (pAttrNames+iAttr)->usID,
357              &((pAttrNames+iAttr)->szName));
358    }
359
360    printf("Ip2ListAttrs ended.\n");
361
362  /*************************************************************************/
```

```
363  /*** Create arguments for Search process                          ***/
364  /*** Perform search                                               ***/
365  /***                                                             ***/
366  /*** Api: SimLibSearch                                           ***/
367  /*****************************************************************/
368
369    printf("Performing SimLibSearch\n");
370
371    printf(" Creating LibSearchCriteriaStruct\n");
372    strcpy(pszSS, "");
373    iCntItemId = 0;
374    iCntSearchArgs = 0;
375
376    for (iSelCrit = 1; iSelCrit <= iCntParmA; iSelCrit++) {
377       pOpPos = strpbrk(pszSelCrit[iSelCrit], "=<>!");
378       if (pOpPos != NULL) {
379          iAttrLen = (int)(pOpPos - pszSelCrit[iSelCrit]);
380          for (iAttr = 0; iAttr < ulCntAttrs; iAttr++) {
381             printf(" iAttr = %i\n", iAttr);
382             if (strnicmp(pszSelCrit[iSelCrit],
383                 (PSZ)&((pAttrNames+iAttr)->szName),
384                 iAttrLen) == 0) {
385                printf (" %s uses A%i.\n", pszSelCrit[iSelCrit],
386                        (pAttrNames+iAttr)->usID);
387                for (iOp = 0; iOp < 4; iOp++) {
388                   printf(" iOp = %i\n", iOp);
389                   if (pszTransOp[iOp] [0] == *pOpPos) {
390                      printf (" Operator %c converted to %s.\n",
391                              *pOpPos,
392                              &pszTransOp[iOp][1]);
393                      break;
394                   }
395                }
396
397                printf (" Value is %s.\n", pOpPos+1);
398                if(iCntSearchArgs > 0) {
399                   strcat(pszSS, " AND");
400                }
401
402                iCntSearchArgs++;
403                sprintf(pszS1,
404                        " A%i %s \"%s\"",
405                        (pAttrNames+iAttr)->usID,
406                        &pszTransOp[iOp][1],
407                        pOpPos+1);
408                strcat (pszSS, pszS1);
409                printf (" pszSS is %s.\n", pszSS);
410
411             }
412          }
413       }
414    }
415
416    pAsyncCtl = NULL;
417    pRC = &RCStruct;
418    usNumCriteria = 1;
419
420    pCriteria->ulReturnLimit = 0;
421    pCriteria->fSearch = SIM_SEARCH_ALLVIEWS;
422    pCriteria->pszSearchString = pszSS;
423
424    printf(" SearchString is %s\n", pCriteria->pszSearchString);
425
```

```
426     SimLibSearch(hSession,
427                  NULL,
428                  NULL,
429                  SIM_SEARCH_STATIC,
430                  SIM_DOCUMENT,
431                  OIM_ALL,
432                  OIM_ALL,
433                  0,
434                  usNumCriteria,
435                  pCriteria,
436                  TRUE,
437                  pAsyncCtl,
438                  pRC);
439
440     if (pRC->ulRC == SIM_RC_OK) {
441        printf(" ulRC from SimLibSearch is SIM_RC_OK.\n");
442     }
443     else {
444        printf(" ulRC from SimLibSearch is %i\n", pRC->ulRC);
445        printf(" ulExtRC from SimLibSearch is %i\n", pRC->ulExtRC);
446        printf(" ulExtReason from SimLibSearch is %i\n", pRC->ulExtReason);
447     }
448
449     printf(" usParam  from SimLibSearch is %p\n", pRC -> usParam );
450     printf(" ulParam1 from SimLibSearch is %p\n", pRC -> ulParam1);
451     printf(" ulParam2 from SimLibSearch is %p\n", pRC -> ulParam2);
452
453     if (pRC -> usParam == 1 &&
454         pRC -> ulParam1 > 0 &&
455         pRC -> ulParam2 > 0) {
456
457        pItemId = (PITEMID)(pRC->ulParam1);
458        iCntItemId = pRC->ulParam2;
459     }
460     printf(" SimLibSearch returned %3d Items\n", iCntItemId);
461     printf("SimLibSearch ended.\n");
462
463    /*********************************************************************/
464    /*** Find output file in DPATH                                     ***/
465    /*** Open output file                                              ***/
466    /*** Copy result set from SimLibSearch to output file              ***/
467    /*** Close output file                                             ***/
468    /*********************************************************************/
469
470     if (iCntItemId > 0) {
471        printf("Searching for output file %s\n", pszNamIp3Result);
472        DosSearchPathRC = DosSearchPath(SEARCH_CUR_DIRECTORY |
473                                        SEARCH_ENVIRONMENT,
474                                        "DPATH",
475                                        pszNamIp3Result,
476                                        pszNamIp3ResultTxt,
477                                        sizeof(pszNamIp3ResultTxt));
478        if (DosSearchPathRC != 0) {
479           strcpy(pszNamIp3ResultTxt, pszNamIp3Result);
480        }
481
482        printf(" Output file is %s\n",pszNamIp3ResultTxt);
483        hIp3ResultTxt = fopen(pszNamIp3ResultTxt,"a");
484        for (iItemId = 0; iItemId <= iCntItemId; iItemId++) {
485           fprintf(hIp3ResultTxt, "%s\n", pItemId+iItemId);
486            printf("Item ID(s) that match search are: %s\n", pItemId+iItemId);
487        }
488
```

```
489     fclose(hIp3ResultTxt);
490   }
491
492   /*********************************************************************/
493   /*** If logon was logal, logoff now.                               ***/
494   /***                                                               ***/
495   /*** Api: SimLibLooff                                              ***/
496   /*********************************************************************/
497
498       printf("Performing logoff\n");
499
500       pRC = &RCStruct;
501       pAsyncCtl = NULL;
502       SimLibLogoff(hSession,
503                    pAsyncCtl,
504                    pRC);
505       printf(" SimLibLogoff complete.\n");
506      if (pRC->ulRC == SIM_RC_OK) {
507        printf(" ulRC from SimLibLoff is SIM_RC_OK.\n");
508      }
509      else {
510        printf(" ulRC from SimLibLogoff is %i\n", pRC->ulRC);
511      }
512
513   /*********************************************************************/
514   /*** End                                                           ***/
515   /*********************************************************************/
516
517     printf("IP3SRCHD Ended\n");
518
519     return(0);
520 }
```

A.4 IP3SRCHD.DEF (11/04/93 15:43:20)

```
1   ;*BEGINPROLOGUE*****************************************************************/
2   ;*                                                                              */
3   ;* MODULE: IP3SRCHD.DEF                                                        */
4   ;*                                                                              */
5   ;* Description: IP3SRCHD.DEF is used by program IP3SRCHD.C.                    */
6   ;*                                                                              */
7   ;* OS          : OS/2 2.1                                                       */
8   ;* Compiler    : C Set/2                                                        */
9   ;*                                                                              */
10  ;*ENDPROLOGUE*****************************************************************/
11
12  NAME IP3SRCHD WINDOWCOMPAT
13
14  DESCRIPTION 'IP/3 Building Blocks - IP3SRCHD'
15
16  STUB    'OS2STUB.EXE'
17
18  CODE    MOVEABLE
19  DATA    MOVEABLE MULTIPLE
20
21  HEAPSIZE   64000        ; Must be non-zero to use Local memory manager
22  STACKSIZE  64000        ; Must be non-zero for SS == DS
23                          ; suggest 4k as minimum stacksize
24  ;*
25  ;* End of IP3SCAN.DEF
```

A.5 IP3SRCHD.MAK (05/18/94 11:24:36)

```
#*****************************************************************************
#
# Make file for IP3SRCHD.C                        ImagePlus VisualInfo
#
#*****************************************************************************
#
#  COPYRIGHT:
#  ----------
#  Copyright (C) International Business Machines Corp., 1993, 1994.
#
#
#  DISCLAIMER OF WARRANTIES:
#  -------------------------
#  The following [enclosed] code is sample code created by IBM
#  Corporation.  This sample code is not part of any standard IBM product
#  and is provided to you solely for the purpose of assisting you in the
#  development of your applications.  The code is provided "AS IS",
#  without warranty of any kind.  IBM shall not be liable for any damages
#  arising out of your use of the sample code, even if they have been
#  advised of the possibility of such damages.
#
#*****************************************************************************
#
# External Environment Variables Used
#
#   SB_ROOT       This environment variable defines where the
#                 VisualInfo directory is located.  It maybe set
#                 either in the config.sys file or from the
#                 command line. For example,
#
#                     SET SB_ROOT=D:\FRNV1R0
#
#   DEBUG         This environment variable defines whether a
#                 Debug or Non-Debug build is to be performed.
#                 It maybe set either in the config.sys file or
#                 from the command line(when switching back and
#                 forth is desired).  The variable may be set
#                 according to
#
#                     SET DEBUG=1  --> perform Debug build
#
#                     or,
#
#                     SET DEBUG=0  --> perform Non-Debug build
#
#*****************************************************************************
#
# Compiler Options Used
#
#  /C+          Compile only
#  /Fd-         Store internal work files in shared memory
#  /Gd+         Dynamically link the run-time library
#  /Ge+         Build an EXE file
#  /Ge-         Build a  DLL file
#  /Gh+         Generate code for profiling
#  /Gh-         Disable profiling
#  /Gm+         Link with multi-threaded version of library
#  /I           Include file location
#  /Kbcfogapexir  Give all diagnostics except preprocessor trace
#  /Lf-         Set all listing options off
#  /Mp          Use - optlink - linkage for functions
```

```
62  #  /Q+          Do not display logo
63  #  /Re          Generate executable code that can be used in an OS/2 runtime
64  #               environment
65  #  /Rn          Generate executable code that can be used as a subsystem with
66  #               no runtime environment
67  #  /Se          Allow all C Set ++ language extensions except migration
68  #  /Sm          Control Compiler interpretation of unsupported keywords
69  #  /Sn+         Allow use of DBCS
70  #  /Sp1         Align on single byte boundires (i.e pack)
71  #  /Sp4         Align on full word boundires (i.e don't pack)
72  #  /Ss          Allow double slash (//) for comments
73  #  /Ti+         Generate debugging information
74  #
75  #  /ALIGN       Set the alignment factor.  Must be power of 2.
76  #  /CO          Include symbolic debugger info
77  #  /DE          Prepare for debugging.
78  #  /EXEPACK     Compress byte pattern in *.exe
79  #  /MAP         Create *.map file.  Lisat public symbols.
80  #  /NOI         NOIGNORECASE           identifiers are case sensitive
81  #  /NOL         NOLOGO                 disable signon banner
82  #  /NOD         NODEFAULTLIBRARYSEARCH don't search LIB libraries
83  #  /PACKCODE    Pack neighboring code segments.
84  #  /PACKDATA    Pack neighboring data segments.
85  #
86  #*****************************************************************************
87
88  #┌"
89  #] Generic/Common Macros                                                    ]
90  #Î┐
91  CCFLAGS  = /C+ /Gd+ /Ge+ /Gm+ /Kb /Mp /Q+ /Re /Se /Sn+ /Sp1 /Ss
92
93  INCLUDES = $(SB_ROOT)\INCLUDE;$(SB_ROOT)\INC
94
95  LLFLAGS  = /NOI /NOL
96
97  #┌"
98  #] Set up defaults                                                         ]
99  #Î┐
100
101 !ifndef DEBUG
102 DEBUG=0
103 !endif
104
105 #┌"
106 #] Debug/Non-Debug Macros                                                  ]
107 #Î┐
108 !if $(DEBUG)
109 CFLAGS   =  /Gh- /Lf- /Ti+ $(CCFLAGS) /I $(INCLUDES)
110 LFLAGS   = /CO /DE $(LLFLAGS)
111 BINDIR   = .
112 LIBDIR   = $(SB_ROOT)\LIB
113 SRCDIR   = .
114 OBJDIR   = .
115 SRCFIL   = IP3SRCHD
116 !else
117 CFLAGS   = $(CCFLAGS) /I $(INCLUDES)
118 LFLAGS   = $(LLFLAGS)
119 BINDIR   = .
120 LIBDIR   = $(SB_ROOT)\LIB
121 SRCDIR   = .
122 OBJDIR   = .
123 SRCFIL   = IP3SRCHD
124 !endif
125
126 COMPILE_REG = ICC $(CFLAGS) -Fo$(OBJDIR)\$@ $(SRCDIR)\$(@B).C
127
```

```
128  LINKLIBS = $(LIBDIR)\FRNOFI.LIB    \
129             $(LIBDIR)\FRNOFO.LIB    \
130             $(LIBDIR)\FIWSENV.LIB   \
131             $(LIBDIR)\FIWSP.LIB     \
132             $(LIBDIR)\FIWSWS.LIB    \
133             $(LIBDIR)\FIWSD.LIB     \
134             $(LIBDIR)\FRNOFI2.LIB   \
135             OS2386.LIB
136
137  OFILES   = $(OBJDIR)\$(SRCFIL).OBJ
138
139  #├"
140  #] Dependencies                                                    ]
141  #Î┐
142
143  FOLDERMGR_DEP = {$(INCLUDES);}FRNP.H         {$(INCLUDES);}FRNPCAPI.H  \
144                  {$(INCLUDES);}FRNPERR.H      {$(INCLUDES);}FRNPFI.H    \
145                  {$(INCLUDES);}FRNPTYPE.H     {$(INCLUDES);}FRNPFI2.H   \
146                  {$(INCLUDES);}FRNPLIBC.H     {$(INCLUDES);}FRNPLCLI.H  \
147                  {$(INCLUDES);}FRNPFO.H
148
149  IS_DEP        = {$(INCLUDES);}FIWS.H         {$(INCLUDES);}FIWSDSP.H   \
150                  {$(INCLUDES);}FIWSWS.H       {$(INCLUDES);}FIWSENV.H   \
151                  {$(INCLUDES);}FIWSPRT.H      {$(INCLUDES);}FIWSDEV.H
152
153  FRNOEXDI_DEP  = {$(INCLUDES);}$(SRCFIL).H     $(FOLDERMGR_DEP)         \
154                   $(IS_DEP)
155
156  #├"
157  #] Main Target Rule                                                ]
158  #Î┐
159  ALL         : ERASE $(BINDIR)\$(SRCFIL).EXE
160
161  ERASE       :
162      if exist ERR.LST erase ERR.LST
163
164  #├"
165  #] Rules                                                           ]
166  #Î┐
167  $(BINDIR)\$(SRCFIL).EXE   :  $(OFILES) $(SRCDIR)\$(SRCFIL).DEF $(SRCFIL).MAK
168     LINK386 $(LFLAGS) $(OFILES),    \
169             $(BINDIR)\$(SRCFIL).EXE, \
170             $(BINDIR)\$(SRCFIL).MAP, \
171             $(LINKLIBS),            \
172             $(SRCDIR)\$(SRCFIL).DEF
173
174  $(OBJDIR)\$(SRCFIL).OBJ : $(SRCDIR)\$(SRCFIL).C $(FRNOEXDI_DEP)
175     $(COMPILE_REG)
```

Appendix B. IP3SCAN Source Code

B.1 Function index

Cleanup (IP3SCAN.C, page 92)	
calls	**SimLibLogoff** **SimWsDeleteObj** **WinDestroyMsgQueue** **WinDestroyWindow** **WinTerminate**
called by	**main** (IP3SCAN.C, page 80)

DisplayObject (IP3SCAN.C, page 85)	
calls	**IpsMessage** (IP3SCAN.C, page 93) **SimDspCreateWin** **SimDspSetWin** **SimOpsInitEnv** **SimWsCreateObj** **strcat** **strcpy**
called by	**main** (IP3SCAN.C, page 80)

DosCreateThread	
called by	**main** (IP3SCAN.C, page 80)

DosExit	
called by	**ScanThread** (IP3SCAN.C, page 88)

DosGetDateTime	
called by	**ScanThread** (IP3SCAN.C, page 88)

IP3DispWinProc1 (IP3SCAN.C, page 83)	
calls	**WinBeginPaint** **WinDefWindowProc** **WinEndPaint** **WinFillRect**

IpsMessage (IP3SCAN.C, page 93)	
calls	**sprintf** **WinMessageBox**
called by	**DisplayObject** (IP3SCAN.C, page 85) **main** (IP3SCAN.C, page 80) **ScanThread** (IP3SCAN.C, page 88)

Logon (IP3SCAN.C, page 84)	
calls	**SimLibLogon**
called by	**main** (IP3SCAN.C, page 80)

main (IP3SCAN.C, page 80)	
calls	**Cleanup** (IP3SCAN.C, page 92) **DisplayObject** (IP3SCAN.C, page 85) **DosCreateThread** **IpsMessage** (IP3SCAN.C, page 93) **Logon** (IP3SCAN.C, page 84) **SimLibFree** **strcat** **strcpy** **strncpy** **strnicmp** **WinCreateMsgQueue** **WinCreateStdWindow** **WinDispatchMsg** **WinGetMsg** **WinInitialize** **WinRegisterClass**

memset	
called by	**ScanThread** (IP3SCAN.C, page 88)

ScanThread (IP3SCAN.C, page 88)	
calls	**DosExit** **DosGetDateTime** **IpsMessage** (IP3SCAN.C, page 93) **memset** **SimLibCreateItem** **SimLibCreateObject** **SimLibFree** **SimOpsFreeMem** **SimScnBasicScan** **SimWsQueryObj** **SimWsStoreObj** **sprintf** **strcat** **strcpy** **strncpy** **WinCreateMsgQueue** **WinDestroyMsgQueue** **WinInitialize** **WinTerminate**

SimDspCreateWin	
called by	**DisplayObject** (IP3SCAN.C, page 85)

SimDspSetWin	
called by	**DisplayObject** (IP3SCAN.C, page 85)

SimLibCreateItem	
called by	**ScanThread** (IP3SCAN.C, page 88)

SimLibCreateObject	
called by	**ScanThread** (IP3SCAN.C, page 88)

SimLibFree	
called by	**main** (IP3SCAN.C, page 80) **ScanThread** (IP3SCAN.C, page 88)

SimLibLogoff	
called by	**Cleanup** (IP3SCAN.C, page 92)

SimLibLogon	
called by	**Logon** (IP3SCAN.C, page 84)

SimOpsFreeMem	
called by	**ScanThread** (IP3SCAN.C, page 88)

SimOpsInitEnv	
called by	**DisplayObject** (IP3SCAN.C, page 85)

SimScnBasicScan	
called by	**ScanThread** (IP3SCAN.C, page 88)

SimWsCreateObj	
called by	**DisplayObject** (IP3SCAN.C, page 85)

SimWsDeleteObj	
called by	**Cleanup** (IP3SCAN.C, page 92)

SimWsQueryObj	
called by	**ScanThread** (IP3SCAN.C, page 88)

SimWsStoreObj	
called by	**ScanThread** (IP3SCAN.C, page 88)

sprintf	
called by	**IpsMessage** (IP3SCAN.C, page 93) **ScanThread** (IP3SCAN.C, page 88)

strcat	
called by	**DisplayObject** (IP3SCAN.C, page 85) **main** (IP3SCAN.C, page 80) **ScanThread** (IP3SCAN.C, page 88)

strcpy	
called by	**DisplayObject** (IP3SCAN.C, page 85) **main** (IP3SCAN.C, page 80) **ScanThread** (IP3SCAN.C, page 88)

strncpy	
called by	**main** (IP3SCAN.C, page 80) **ScanThread** (IP3SCAN.C, page 88)

strnicmp	
called by	**main** (IP3SCAN.C, page 80)

WinBeginPaint	
called by	**IP3DispWinProc1** (IP3SCAN.C, page 83)

WinCreateMsgQueue	
called by	**main** (IP3SCAN.C, page 80) **ScanThread** (IP3SCAN.C, page 88)

WinCreateStdWindow	
called by	**main** (IP3SCAN.C, page 80)

WinDefWindowProc	
called by	**IP3DispWinProc1** (IP3SCAN.C, page 83)

WinDestroyMsgQueue	
called by	**Cleanup** (IP3SCAN.C, page 92) **ScanThread** (IP3SCAN.C, page 88)

WinGetMsg	
called by	**main** (IP3SCAN.C, page 80)

WinDestroyWindow	
called by	**Cleanup** (IP3SCAN.C, page 92)

WinInitialize	
called by	**main** (IP3SCAN.C, page 80) **ScanThread** (IP3SCAN.C, page 88)

WinDispatchMsg	
called by	**main** (IP3SCAN.C, page 80)

WinMessageBox	
called by	**IpsMessage** (IP3SCAN.C, page 93)

WinEndPaint	
called by	**IP3DispWinProc1** (IP3SCAN.C, page 83)

WinRegisterClass	
called by	**main** (IP3SCAN.C, page 80)

WinFillRect	
called by	**IP3DispWinProc1** (IP3SCAN.C, page 83)

WinTerminate	
called by	**Cleanup** (IP3SCAN.C, page 92) **ScanThread** (IP3SCAN.C, page 88)

B.2 IP3SCAN.H (05/20/94 14:50:20)

```
1  /*****************************************************************************/
2  /*                                                                           */
3  /* MODULE: IP3SCAN.H                                                         */
4  /*                                                                           */
5  /* Description  : Header file for module IP3SCAN.C.                          */
6  /*                                                                           */
7  /*****************************************************************************/
8
9  /*****************************************************************************/
10 /* IP3SCAN.C Function Prototypes                                             */
11 /*****************************************************************************/
12
13 ULONG    Logon              (PRCSTRUCT pRC,
14                              PSZ pszDBName,
15                              PSZ pszApplicationName,
16                              PSZ pszUserID,
17                              PSZ pszPassword) ;
18 ULONG    DisplayObject       (PRCSTRUCT pRC) ;
19 extern VOID ScanThread       (VOID) ;
20 VOID     Cleanup            (PRCSTRUCT pRC) ;
21 VOID     IpsMessage          (PSZ   pszMsgText,
22                              ULONG ulRC) ;
23
24 /*****************************************************************************/
25 /* SimLibLogon Parameter Definitions                                         */
26 /*****************************************************************************/
27
28 #define PSZDBNAME          "LIBSRVR2"
29 #define PSZAPPLICATIONNAME "SAMPLE"
30 #define PSZUSERID          "FRNADMIN"
31 #define PSZPASSWORD        "PASSWORD"
32
```

```
/****************************************/
/* SimLibLogon Parameter Maximum Lengths */
/****************************************/

#define MAXDBNAME           18          /* Maximum size of pszDBName         */
#define MAXAPPLICATIONNAME  8           /* Maximum size of pszApplicationName*/
#define MAXUSERID           26          /* Maximum size of pszUserID         */
#define MAXPASSWORD         8           /* Maximum size of pszPassword       */

/*****************************************************************************/
/* SimOpsInitEnv Parameter Definitions                                       */
/*****************************************************************************/

#define PSZAPPLDESC         "Sample scenario to print a document."
#define PSZWORKINGDIR       "D:\\FRNV1RO"
#define PSZEXITDIR          "D:\\FRNV1RO\\DLL"
#define PSZOIMANAGER        "FRNOFI"
#define PSZOMANAGER         "FRNOFO"

/*****************************************/
/* SimOpsInitEnv Parameter Maximum Lengths */
/*****************************************/

#define MAXDIRNAME           255        /* Maximum size of pszWorkingDir,    */
                                        /* pszExitDir, pszOIManager,         */
                                        /* and pszOManager                   */
#define MAXDESC              40         /* Maximum size of pszApplDesc       */
                                        /* and pszDocDes                     */

/*****************************************/
/* Message Texts                         */
/*****************************************/

#define PSZMSGPARMERR        "Parameter Unrecognized, Parm # "
#define PSZMSGPRTFAIL        "Print Document Error.  RC ="
#define PSZMSGLOGONFAIL      "Library Logon error.  RC ="
#define PSZMSGOPSINITFAIL    "SimOpsInitEnv error.  RC ="
#define PSZMSGWSCREATEOBJFAIL  "SimWsCreateObj error.  RC ="
#define PSZMSGDISPFAIL       "Display Object error.  RC ="
#define PSZMSGSCANTHREADFAIL "Error starting thread for Scan.  RC ="
#define PSZMSGQUERYOBJFAIL1  "SimWsQueryObj (NUM_CHILD) error.  RC ="
#define PSZMSGQUERYOBJFAIL2  "SimWsQueryObj (LIST_CHILD) error.  RC ="
#define PSZMSGNOSCANOBJ      "No objects were scanned.  RC ="
#define PSZMSGBASICSCANFAIL  "SimScnBasicScan error.  RC ="
#define PSZMSGCREATEOBJFAIL  "SimLibCreateObj error.  RC ="
#define PSZMSGSTOREOBJFAIL   "SimWsStoreObj error.  RC ="
#define PSZMSGCREATEITEMFAIL "SimLibCreateItem error.  RC ="
#define PSZMSGCREATEITEMOK   " Document Item created.  RC ="
#define PSZMSGDISPOBJECTFAIL "SimDspCreateWin error.  RC ="
#define PSZMSGDISPSETDILFAIL "SimDspSetWin (set DIL) error.  RC ="
#define PSZMSGDISPSHOWFAIL   "SimDspSetWin (show window) error.  RC ="
#define PSZMSGDISPUPDTFAIL   "SimDspSetWin (set update) error.  RC ="

#define PSZTITLEBARTEXT      "IP3Scan New Document"

/*****************************************************************************/
/* PM Stuff                                                                  */
/*****************************************************************************/

#define IDFRAME 100
MRESULT EXPENTRY IP3DispWinProc1(HWND hwnd, ULONG msg, MPARAM mp1, MPARAM mp2);

/*****************************************************************************/
/* End of IP3SCAN.H                                                          */
/*****************************************************************************/
```

B.3 IP3SCAN.C (06/28/94 14:05:42)

Cleanup (IP3SCAN.C, page 92)	
calls	**SimLibLogoff** **SimWsDeleteObj** **WinDestroyMsgQueue** **WinDestroyWindow** **WinTerminate**
called by	**main** (IP3SCAN.C, page 80)

DisplayObject (IP3SCAN.C, page 85)	
calls	**IpsMessage** (IP3SCAN.C, page 93) **SimDspCreateWin** **SimDspSetWin** **SimOpsInitEnv** **SimWsCreateObj** **strcat** **strcpy**
called by	**main** (IP3SCAN.C, page 80)

IP3DispWinProc1 (IP3SCAN.C, page 83)	
calls	**WinBeginPaint** **WinDefWindowProc** **WinEndPaint** **WinFillRect**

IpsMessage (IP3SCAN.C, page 93)	
calls	**sprintf** **WinMessageBox**
called by	**DisplayObject** (IP3SCAN.C, page 85) **main** (IP3SCAN.C, page 80) **ScanThread** (IP3SCAN.C, page 88)

Logon (IP3SCAN.C, page 84)	
calls	**SimLibLogon**
called by	**main** (IP3SCAN.C, page 80)

main (IP3SCAN.C, page 80)	
calls	**Cleanup** (IP3SCAN.C, page 92) **DisplayObject** (IP3SCAN.C, page 85) **DosCreateThread** **IpsMessage** (IP3SCAN.C, page 93) **Logon** (IP3SCAN.C, page 84) **SimLibFree** **strcat** **strcpy** **strncpy** **strnicmp** **WinCreateMsgQueue** **WinCreateStdWindow** **WinDispatchMsg** **WinGetMsg** **WinInitialize** **WinRegisterClass**

ScanThread (IP3SCAN.C, page 88)	
calls	**DosExit** **DosGetDateTime** **IpsMessage** (IP3SCAN.C, page 93) **memset** **SimLibCreateItem** **SimLibCreateObject** **SimLibFree** **SimOpsFreeMem** **SimScnBasicScan** **SimWsQueryObj** **SimWsStoreObj** **sprintf** **strcat** **strcpy** **strncpy** **WinCreateMsgQueue** **WinDestroyMsgQueue** **WinInitialize** **WinTerminate**

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/* MODULE: IP3SCAN.C                                                                */
/*                                                                                  */
/* DISCLAIMER OF WARRANTIES:                                                        */
/* -------------------------                                                        */
/* The following [enclosed] code is sample code created by IBM                      */
/* Corporation.  This sample code is not part of any standard IBM product           */
/* and is provided to you solely for the purpose of assisting you in the            */
/* development of your applications.  The code is provided "AS IS",                 */
/* without warranty of any kind.  IBM shall not be liable for any damages           */
/* arising out of your use of the sample code, even if they have been               */
/* advised of the possibility of such damages.                                      */
/*                                                                                  */
/* Description : IP3SCAN.C - Sample application code to display a document.         */
/*               Folder Manager APIs used are : SimLibLogon, SimLibFree,            */
/*               SimLibLogoff, SimLibCreateItem, SimLibCreateObject.                */
/*               Image Services APIs used are: SimOpsInitEnv,                       */
/*               SimDspCreateWin, SimDspSetWin, SimScnBasicScan,                    */
/*               SimWsCreateObj, SimWsQueryObj, SimWsStoreObj, SimOpsFreeMem.       */
/*                                                                                  */
/* System      : OS/2 compatible PC                                                 */
/* OS          : OS/2 2.1                                                           */
/* Compiler    : IBM C/C++                                                          */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

/************************************************************************************/
/*  System Header Files                                                             */
/************************************************************************************/
#define INCL_WIN
#define INCL_DOSPROCESS
#include <os2.h>                        /* Main OS/2 header files                   */
#include <stdio.h>                      /* Standard I/O header files                */
#include <stdlib.h>                     /* Standard Library header files            */
#include <string.h>                     /* String manipulation                      */

/************************************************************************************/
/*  ImagePlus - Folder Manager header files                                         */
/************************************************************************************/
#define   FRN_INCL_FI                   /* Include frnpfi.h & frnplcli.h            */
#define   FRN_INCL_FO                   /* Include frnpfo.h                         */
#include "frnp.h"                       /* Global types                             */
#include "frnpcapi.h"                   /* Structures and constants                 */

/************************************************************************************/
/*  ImagePlus - Image Services Header Files                                         */
/************************************************************************************/
#define   FIWS_SERVER                   /* Required to read from a server           */
#include "fiws.h"                       /* Base Image Services                      */
#include "fiwsenv.h"                    /* Environment prototypes                   */
#include "fiwsws.h"                     /* Working Set prototypes                   */
#include "fiwsdsp.h"                    /* Display Prototypes                       */
#include "fiwsscn.h"                    /* Scan prototypes                          */

/************************************************************************************/
/*  IP3SCAN Header File                                                             */
/************************************************************************************/
#include "IP3SCAN.h"                    /* Application header file                  */

/************************************************************************************/
/*  Global Variables                                                                */
/************************************************************************************/
/*  OS/2 and PM APIs                                                                */
/************************************************************************************/
HAB       hab   = 0;                    /* Anchor block handle                      */
```

```
67  HMQ       hmq   = 0;                    /* Message Queue handle             */
68  HWND      hwnd  = 0;                    /* Main Window handle               */
69  HWND      hwndc = 0;                    /* handle to client area            */
70  QMSG      qmsg;                         /* Message Queue                    */
71  ULONG     ulCreateFlags;                /* Window frame controls            */
72  TID       tidScan;                      /* For ScanThread                   */
73
74  /*********************************************************************************/
75  /*  ImagePlus APIs - Library Session data used with Folder Manager APIs      */
76  /*********************************************************************************/
77  HSESSION hSession = 0;                  /* Folder Manager session handle    */
78  RCSTRUCT RCStruct;                      /* RC data structure for API calls  */
79  ULONG    ulMemoryBlock;                 /* Generic pointer for SimLibFree   */
80  CHAR     DBName[MAXDBNAME + 1] = "";               /* Library Server Name   */
81  CHAR     ApplicationName[MAXAPPLICATIONNAME + 1] = ""; /* Application Name  */
82  CHAR     UserID[MAXUSERID + 1] = "";               /* User ID               */
83  CHAR     Password[MAXPASSWORD + 1] = "";           /* Password              */
84  ITEMID   DocItemID = "";                /* Document to Scan                 */
85  ULONG    ulObjConCls;                   /* Document content class           */
86
87  /*********************************************************************************/
88  /*  ImagePlus APIs - Image Services Data - used with Image Services APIs     */
89  /*********************************************************************************/
90  SIMWSOBJ hWsObj = 0;                    /* Working set object handle        */
91  SIMWIN    hWin  = 0;                    /* Display window handle            */
92
93  /*********************************************************************************/
94  /*  Define Parameters for SimWsCreateObj                                     */
95  /*********************************************************************************/
96
97  SIMCONTROL_BASE    pControl;            /* Object load control information  */
98
99  /*********************************************************************************/
100 /*  Define Parameters for SimDspCreateWin and SimDspSetWin                   */
101 /*********************************************************************************/
102
103 CHAR          szTitle[40];              /* Display window title             */
104 ULONG         ulCreateFlags;            /* Window frame controls            */
105 SIMPOSSIZE    PosSize;                  /* Initial window position, size    */
106 BOOL          fSimDsp = TRUE;           /* Window Visibility Flag           */
107 SIMUPDATE     ulWindowUpdate;           /* Window Update trigger flag       */
108 PSZ           pszDilFmt =               /* define Dynamic Information Line  */
109               "Page %2PageInDoc. of %2NumPageInDoc. of Document %2DocInWs.";
110
```

```
111  /*BEGINPROLOGUE*********************************************************/
112  /*                                                                      */
113  /*  Function         : main                                             */
114  /*                                                                      */
115  /*  Description      : main() function of sample program to scan a doc. */
116  /*                                                                      */
117  /*  Function type    :  INT.                                            */
118  /*                                                                      */
119  /*  Input Parameters :  /D= ItemID of document to display               */
120  /*                      /N= ApplicationName                             */
121  /*                      /U= UserID                                      */
122  /*                      /P= Password                                    */
123  /*                      /S= DBName (Library Server name)                */
124  /*                                                                      */
125  /*  Output Parameters:  None.                                           */
126  /*                                                                      */
127  /*  Synopsis         :  INT main ( int argc, char *argv[] )             */
128  /*                                                                      */
129  /*  Return values    :  If all functions are successful, the return code*/
130  /*                      from Logoff() is returned.  Otherwise, the return*/
131  /*                      code from the failing function is returned.     */
132  /*                                                                      */
133  /*  Process Flow     :                                                  */
134  /*      1.  Initialize Presentation Manager.                            */
135  /*      2.  Establish a session with the Folder Manager                 */
136  /*      3.  Create a PM Message Queue and a window.                     */
137  /*      4.  Create a display window to display the scanned image.       */
138  /*      5.  Start a thread to perform the scan.                         */
139  /*      6.  Process the message loop to wait for user to quit.          */
140  /*      7.  Free the working set.                                       */
141  /*      8.  Logoff from the Folder Manager session.                     */
142  /*      9.  Exit main() with a return code.                             */
143  /*                                                                      */
144  /*ENDPROLOGUE***********************************************************/
145
146  INT main (int argc, char *argv[]) {
147
148      ULONG  ulretcode   = 0;
149      USHORT i;
150      CHAR   msgText[80] = "";            /* Work string for building msg text */
151
152      /************************************************************************/
153      /* Initialize PM, and create Message Queue so we can use WinMessageBox.  */
154      /* PM is required for Image Services "Display" and "Scan" functions      */
155      /************************************************************************/
156
157      hab = WinInitialize((USHORT) NULL);
158      hmq = WinCreateMsgQueue( hab, 0 );
159      if (hab == NULLHANDLE) {
160          return(99);                                 /* Quit Immediately      */
161      }
162
163      /************************************************************************/
164      /* Initialize Session Variables required for logon.                     */
165      /************************************************************************/
166      strcpy (DBName,           PSZDBNAME);
167      strcpy (ApplicationName,  PSZAPPLICATIONNAME);
168      strcpy (UserID,           PSZUSERID);
169      strcpy (Password,         PSZPASSWORD);
170
171      /************************************************************************/
172      /* Check that Document ID has been passed as parameter.                 */
173      /************************************************************************/
174
175      for (i=1; i < argc; i++) {
```

```
176 |   |   if (strnicmp(argv[i],"/N=",3)==0) {
177 |   |       strncpy( ApplicationName, argv[i] + 3, MAXAPPLICATIONNAME );
178 |   |   }
179 |   |   else if (strnicmp(argv[i],"/U=",3)==0) {
180 |   |       strncpy( UserID,          argv[i] + 3, MAXUSERID          );
181 |   |   }
182 |   |   else if (strnicmp(argv[i],"/S=",3)==0) {
183 |   |       strncpy( DBName,          argv[i] + 3, MAXDBNAME          );
184 |   |   }
185 |   |   else if (strnicmp(argv[i],"/P=",3)==0) {
186 |   |       strncpy( Password,        argv[i] + 3, MAXPASSWORD        );
187 |   |   }
188 |   |   else {
189 |   |       IpsMessage(strcat(strcpy(msgText, argv[i]), PSZMSGPARMERR), i );
190 |   |       Cleanup( &RCStruct );
191 |   |       return(99);                           /* Quit Immediately      */
192 |   |   }
193 |   }
194 |
195 |   /*************************************************************************/
196 |   /*  Call Logon to establish a Folder Manager Session.                    */
197 |   /* If successful, call SimLibFree to free the memory for structure       */
198 |   /* USERLOGONINFOSTRUC returned from SimLibLogon.                         */
199 |   /*************************************************************************/
200 |
201 |   ulretcode = Logon ( &RCStruct,                 /* API Return Code struct*/
202 |                       DBName,                   /* Library Server name   */
203 |                       ApplicationName,          /* Application Name      */
204 |                       UserID,                   /* User ID               */
205 |                       Password);                /* Password              */
206 |
207 |   if (ulretcode == SIM_RC_OK) {
208 |   |
209 |   |   SimLibFree( hSession,                  /* Folder Mgr session handle  */
210 |   |               (PVOID) ulMemoryBlock,     /* Ptr to memory block to free*/
211 |   |               (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure   */
212 |   }
213 |   else {
214 |   |   IpsMessage(PSZMSGLOGONFAIL, ulretcode);
215 |   |   Cleanup( &RCStruct );
216 |   |   return(ulretcode);                        /* Quit Immediately      */
217 |   }
218 |
219 |
220 |   /*************************************************************************/
221 |   /* Create a parent window for IP3SCAN.                                   */
222 |   /*************************************************************************/
223 |
224 |   WinRegisterClass (hab,
225 |                     "IP3ScanMainWindow",
226 |                     (PFNWP) IP3DispWinProc1,
227 |                     CS_SIZEREDRAW,
228 |                     0);
229 |   ulCreateFlags = FCF_TITLEBAR      |
230 |                   FCF_SYSMENU       |
231 |                   FCF_SIZEBORDER    |
232 |                   FCF_MINMAX        |
233 |                   FCF_SHELLPOSITION |
234 |                   FCF_TASKLIST;
235 |   hwnd = WinCreateStdWindow (HWND_DESKTOP,
236 |                              WS_VISIBLE,
237 |                              &ulCreateFlags,
238 |                              "IP3ScanMainWindow",
```

```
                                "IP3Scan Main Window",
                                0L,
                                (HMODULE) NULL,
                                IDFRAME,
                                (PHWND) &hwndc);

    /*************************************************************************/
    /* Call DisplayObject to call Image Services to open a display window.   */
    /*************************************************************************/

    if ( (ulretcode = DisplayObject( &RCStruct) ) != SIM_RC_OK) {
        IpsMessage(PSZMSGDISPFAIL, ulretcode);
        Cleanup( &RCStruct );
        return(ulretcode);                          /* Quit Immediately      */
    }

    /*************************************************************************/
    /* Start a new thread to perform a basic scan into the empty WS object.  */
    /*************************************************************************/

    if ( (ulretcode = DosCreateThread( &tidScan,
                                       (PFNTHREAD) ScanThread,
                                       0,
                                       0,
                                       32768 ) ) != 0) {
        IpsMessage(PSZMSGSCANTHREADFAIL, ulretcode);
        Cleanup( &RCStruct );
        return(ulretcode);                          /* Quit Immediately      */
    }

    /*************************************************************************/
    /* Drop into the PM message loop.                                        */
    /*************************************************************************/
    /*************************************************************************/
    /*   Get and Dispatch messages from the application message queue until  */
    /*   WinGetMsg returns FALSE, indicating a WM_QUIT message.              */
    /*************************************************************************/

    while (WinGetMsg (hab,
                      &qmsg,
                      (HWND) NULL,
                      0,
                      0 )) {
        WinDispatchMsg (hab,
                        &qmsg );
    }

    /*************************************************************************/
    /*   Clean house and terminate the application                           */
    /*************************************************************************/

    Cleanup( &RCStruct );

    return ( ulretcode );
}
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : IP3DispWinProc1                                             */
/*                                                                                 */
/*  Description      : Window proc for dummy window displayed after document       */
/*                     is displayed.  This provides the means to terminate         */
/*                     IP3SCAN after the Image Services Display is complete.       */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

MRESULT EXPENTRY IP3DispWinProc1 (HWND hwnd, ULONG msg, MPARAM mp1, MPARAM mp2) {

    HPS hps;
    RECTL rectl;

    switch ( msg ) {
        case WM_CREATE:
           break;

        case WM_PAINT:
            hps = WinBeginPaint(hwnd, 0L, (PRECTL) &rectl);
            WinFillRect(hps, (PRECTL) &rectl, SYSCLR_WINDOW );
            WinEndPaint(hps);
            break;

       case WM_COMMAND:
            break;

       default:
           return(WinDefWindowProc (hwnd, msg, mp1, mp2));
    }
    return(FALSE);
}

```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : Logon                                                       */
/*                                                                                 */
/*  Description      : This function is called from main() to logon to             */
/*                     ImagePlus Bid Folder Manager.                               */
/*                                                                                 */
/*  Function type    :  ULONG                                                      */
/*                                                                                 */
/*  Input Parameters :  pszDBName pszApplicationName pszUserID pszPassword         */
/*                                                                                 */
/*  Output Parameters:  hSession RCSTRUCT                                          */
/*                                                                                 */
/*  Synopsis         :  ULONG Logon ( pRC                                          */
/*                                    pszDBName,                                   */
/*                                    pszApplicationName,                          */
/*                                    pszUserID,                                   */
/*                                    pszPassword)                                 */
/*                                                                                 */
/*  Return values    :  ulRC return code from SimLibLogon call                     */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*      1.  Initialize SimLibLogon parameters.                                     */
/*      2.  Establish a session with the Folder Manager                            */
/*          by calling SimLibLogon.                                                */
/*      3.  Return to main() with Session Handle                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG Logon (PRCSTRUCT pRC, PSZ pszDBName,
             PSZ pszApplicationName, PSZ pszUserID, PSZ pszPassword) {

    /*************************************************************************/
    /* Initialize SimLibLogon parameters.                                    */
    /*************************************************************************/

    RCStruct.ulRC      = SIM_RC_OK;

    /*************************************************************************/
    /* Call SimLibLogon to establish a Folder Manager session.               */
    /* If successful, save the Folder Manager session handle for subsequent  */
    /* Folder Manager API calls.                                             */
    /*************************************************************************/

    SimLibLogon( (PSZ) DBName,            /* Pointer to Database name        */
                 (PSZ) ApplicationName,   /* Pointer to Application Name     */
                 (PSZ) UserID,            /* Pointer to User Id              */
                 (PSZ) Password,          /* Pointer to Password for User Id */
                 (PSZ) NULL,              /* New Password */
                 (PSZ) NULL,              /* Logon without proxy             */
                 (PSZ) NULL,              /* Logon without proxy scope       */
                 SIM_SS_NORMAL,           /* Non-configuration session logon */
                 (PASYNCCTLSTRUCT) NULL,  /* Request synchronous processing  */
                 (PRCSTRUCT) &RCStruct ); /* Pointer to RC data structure    */

    if ( RCStruct.ulRC == SIM_RC_OK) {

        hSession = RCStruct.ulParam1;              /* Save Session Handle   */
                                                   /* Save ptr to USERLOGONINFOSTRUCT */
        ulMemoryBlock = RCStruct.ulParam2;         /* for use with SimLibFree */
    }
    return( RCStruct.ulRC );
}
```

```
393  /*BEGINPROLOGUE*****************************************************************/
394  /*                                                                               */
395  /*  Function         : DisplayObject                                             */
396  /*                                                                               */
397  /*  Description      : This function is called from main() to perform the        */
398  /*                     functions necessary to display an (initially) empty       */
399  /*                     Working Set object.                                       */
400  /*                                                                               */
401  /*  Function type    :  ULONG.                                                   */
402  /*                                                                               */
403  /*  Input Parameters :  None                                                     */
404  /*                                                                               */
405  /*  Output Parameters:  None.                                                    */
406  /*                                                                               */
407  /*  Synopsis         :  ULONG DisplayObject ( &RCSTRUCT )                        */
408  /*                                                                               */
409  /*  Return values    :  If all functions are successful, SIM_RC_OK is            */
410  /*                      returned.  Otherwise the return code from                */
411  /*                      SimOpsInitEnv, SimWsCreateObj, SimDspCreateWin,          */
412  /*                      or SimDspSetWin is returned.                             */
413  /*                                                                               */
414  /*  Process Flow     :                                                           */
415  /*      1.  Initialize the Image Services Environment.                           */
416  /*      2.  Create an empty WS object as the eventual target for scan.           */
417  /*      3.  Display the Empty WS object.                                         */
418  /*      4.  Return to main() with a return code.                                 */
419  /*                                                                               */
420  /*ENDPROLOGUE*******************************************************************/
421
422  ULONG DisplayObject ( PRCSTRUCT pRC ) {
423
424      /*************************************************************************/
425      /*  Make sure RCStruct initialized properly                              */
426      /*************************************************************************/
427
428      RCStruct.ulRC     = SIM_RC_OK;
429      RCStruct.ulStruct = sizeof(RCSTRUCT);    /* must initialize length field */
430
431      /*************************************************************************/
432      /* Call SimOpsInitEnv to initialize the environment.                     */
433      /*   - parameter values defined in IP3PRINT.H                            */
434      /*************************************************************************/
435
436      if (SimOpsInitEnv(PSZAPPLDESC,       /* Application description          */
437                  FALSE,                  /* Is this instance unattended?     */
438                  PSZWORKINGDIR,          /* Working directory for application*/
439                  PSZEXITDIR,             /* User exit directory              */
440                  NULL,
441                  PSZOMANAGER,            /* Folder Mgr Object Manager DLL    */
442                  NULL,
443                  (PRCSTRUCT) &RCStruct )   /* Pointer to RC data structure  */
444      !=          SIM_RC_OK ) {
445          IpsMessage(PSZMSGOPSINITFAIL, RCStruct.ulRC);  /* Display Error Msg  */
446          return( RCStruct.ulRC );
447      }
448
449      /*************************************************************************/
450      /* Initialize SimWsCreateObj parameters                                  */
451      /*************************************************************************/
452
453      /***************************************/
454      /* SIMCONTROL_BASE structure.          */
455      /***************************************/
456
457      pControl.ulStruct     = sizeof(SIMCONTROL_BASE);
458      pControl.fReserved    = FALSE;
```

```
459     pControl.pszDesc      = NULL;
460
461     /***********************************/
462     /* All attributes are visible.     */
463     /***********************************/
464
465     pControl.ulHideFrom   = SIMUSER_NONE;
466     pControl.ulDocControl = 0;
467
468     /*****************************************************************************/
469     /* Call SimWsCreateObj to create a WS object to place scanned document.  */
470     /*****************************************************************************/
471
472     if (SimWsCreateObj(SIMREF_NONE,   /* Create standalone WS Object         */
473           NULL,                      /* Handle to WS Object (ignored)       */
474           (PVOID)&pControl,          /* Load control information            */
475           (PSIMWSOBJ)&hWsObj,        /* Handle of working set object created*/
476           (PRCSTRUCT) &RCStruct )    /* Pointer to RC data structure        */
477   ┌               != SIM_RC_OK ) {
478   │     IpsMessage(PSZMSGWSCREATEOBJFAIL, RCStruct.ulRC); /* Display Error   */
479   │     return( RCStruct.ulRC );
480   └ }
481
482     /*****************************************************************************/
483     /* Initialize SimDspCreateWin parameters.                                    */
484     /*****************************************************************************/
485
486     /*****************************************************************************/
487     /* Create the window with all standard frame features.                       */
488     /*****************************************************************************/
489
490     ulCreateFlags = SIMDSP_FRAME_BORDER |
491                     SIMDSP_FRAME_MINMAX |
492                     SIMDSP_FRAME_RESIZE |
493                     SIMDSP_FRAME_SYSTEM |
494                     SIMDSP_FRAME_TITLE;
495
496     /*****************************************************************************/
497     /* Specify the window origin.  The origin is taken from the top/left.    */
498     /*****************************************************************************/
499
500     PosSize.ulStruct               =  sizeof(SIMPOSSIZE);
501     PosSize.uomType                =  SIMUOM_PIXEL;
502     PosSize.ratOriginX.lNumerator  =  0L;
503     PosSize.ratOriginX.lDenominator =  1;
504     PosSize.ratOriginY.lNumerator  =  0L;
505     PosSize.ratOriginY.lDenominator =  1;
506     PosSize.ratWidth.lNumerator    =  400;
507     PosSize.ratWidth.lDenominator  =  1;
508     PosSize.ratHeight.lNumerator   =  600;
509     PosSize.ratHeight.lDenominator  =  1;
510
511     /*****************************************************************************/
512     /* Setup window title bar.                                                   */
513     /*****************************************************************************/
514
515     strcpy (szTitle, PSZTITLEBARTEXT);
516     strcat (szTitle, DocItemID);
517
518     /*****************************************************************************/
519     /* Call SimDspCreateWin to create a new display window in order to       */
520     /* view the working set data.                                                */
521     /*****************************************************************************/
522
523     if (SimDspCreateWin(
```

```
524           HWND_DESKTOP,               /* Parent window is the desktop       */
525           HWND_DESKTOP,               /* Owner window is the desktop        */
526           0,
527           szTitle,                    /* Display window title               */
528           (SIMWSOBJ)hWsObj,           /* Working set object handle          */
529           (PSIMPOSSIZE)&PosSize,      /* Window's initial position and size */
530           (ULONG)ulCreateFlags,       /* Window frame controls              */
531           FALSE,                      /* Create window invisible            */
532           (PSIMWIN)&hWin,             /* Pointer to display window handle   */
533           (PRCSTRUCT) &RCStruct )     /* Pointer to RC data structure       */
534       ┌               != SIM_RC_OK ) {
535       │   IpsMessage(PSZMSGDISPOBJECTFAIL, RCStruct.ulRC);  /* Display Err Msg */
536       └}
537
538       /*************************************************************************/
539       /* Define the Dynamic Information Line for the Display Window            */
540       /*************************************************************************/
541
542       if (SimDspSetWin( hWin,
543                         SIMDSP_DIL,
544                         (PVOID) &pszDilFmt,
545                         &RCStruct )
546       ┌               != SIM_RC_OK ) {
547       │   IpsMessage(PSZMSGDISPSETDILFAIL, RCStruct.ulRC);  /* Display Err Msg */
548       └}
549       if (SimDspSetWin( hWin,
550                         SIMDSP_SHOW,
551                         (PVOID) &fSimDsp,
552                         &RCStruct )
553       ┌               != SIM_RC_OK ) {
554       │   IpsMessage(PSZMSGDISPSHOWFAIL, RCStruct.ulRC);    /* Display Err Msg */
555       └}
556       ulWindowUpdate = SIMUPDATE_EVERY_PAGE;
557       if (SimDspSetWin( hWin,
558                         SIMDSP_UPDATE,
559                         (PVOID) &ulWindowUpdate,
560                         &RCStruct )
561       ┌               != SIM_RC_OK ) {
562       │   IpsMessage(PSZMSGDISPUPDTFAIL, RCStruct.ulRC);    /* Display Err Msg */
563       └}
564
565       return( RCStruct.ulRC );
566   └}
567
```

```
568  /*BEGINPROLOGUE*****************************************************************/
569  /*                                                                               */
570  /*  Function         : ScanThread                                                */
571  /*                                                                               */
572  /*  Description      : This function is called from main() to perform the        */
573  /*                     Scan.                                                     */
574  /*                                                                               */
575  /*  Function type    :  VOID.                                                    */
576  /*                                                                               */
577  /*  Input Parameters :  None                                                     */
578  /*                                                                               */
579  /*  Output Parameters:  None.                                                    */
580  /*                                                                               */
581  /*  Synopsis         :  VOID ScanThread (VOID)                                   */
582  /*                                                                               */
583  /*  Return values    :  If all functions are successful, SIM_RC_OK is            */
584  /*                      returned.  Otherwise the return code from                */
585  /*                      SimOpsInitEnv, SimWsCreateObj, SimDspCreateWin,          */
586  /*                      or SimDspSetWin is returned.                             */
587  /*                                                                               */
588  /*  Process Flow     :                                                           */
589  /*      1.  Initialize PM environment for this thread                            */
590  /*      2.  Call SimScnBasicScan to scan the document int the Working Set        */
591  /*      3.  Query WS to check that a document has been scanned                   */
592  /*      4.  Create a new document item to store the scanned document             */
593  /*      5.  Create a new object in this document to contain the scanned doc      */
594  /*      6.  Store the document from WS into this library object                  */
595  /*      7.  Return to main() with a return code.                                 */
596  /*                                                                               */
597  /*ENDPROLOGUE*******************************************************************/
598
599  VOID  ScanThread ( VOID ) {
600
601      PSIMWSOBJ pScanObj=NULL;            /* Handle to scanned document        */
602      ULONG     ulNumObj=0;               /* Number of objects in working set  */
603      USHORT    count;                    /* Loop counter                      */
604      OBJ       Obj;                      /* For SimLibCreateObject            */
605      OBJ       Store;                    /* For SimWsStoreObj                 */
606      SIMSRCSERVEROBJ DataSrc;            /* Source from which to load data    */
607      RCSTRUCT        rcStruct;           /* RC data structure                 */
608      ATTRLISTSTRUCT  Docu [3];                   /* Document Attributes       */
609      CHAR            szTimeStamp[26];            /* String for Time Stamp     */
610      DATETIME        DateTime;                   /* For System Date/Time info */
611      CHAR            msgText[80]  = "";  /* Work string for building msg text */
612
613      HAB   habscan = 0;                  /* Anchor block handle for thread    */
614      HMQ   hmqscan = 0;                  /* Message Queue handle for thread   */
615
616      rcStruct.ulStruct = sizeof(RCSTRUCT);
617
618      /*************************************************************************/
619      /* Initialize PM, and create Message Queue so we can use WinMessageBox.  */
620      /* from this thread.                                                     */
621      /*************************************************************************/
622
623      habscan = WinInitialize((USHORT) NULL);
624      hmqscan = WinCreateMsgQueue( hab, 0 );
625      if (habscan == NULLHANDLE) {
626          DosExit(EXIT_THREAD, 99);                   /* Quit immediately      */
627      }
628
629      /*************************************************************************/
630      /* Do the Scan.  Since this is synchronous, return means scan complete.  */
631      /*************************************************************************/
632
```

```
633     if (SimScnBasicScan ( hWin,
634                           (PSZ) ApplicationName, NULL, NULL,
635         &rcStruct ) != SIM_RC_OK) {
636         IpsMessage(PSZMSGBASICSCANFAIL, RCStruct.ulRC);   /* Display Err Msg */
637         DosExit(EXIT_THREAD, RCStruct.ulRC);
638     }
639
640     /*************************************************************************/
641     /* Check that the scan has actually created some object(s)               */
642     /*************************************************************************/
643
644     if (SimWsQueryObj((SIMWSOBJ)hWsObj,
645                       SIMUSER_NONE,
646                       SIMWS_NUM_CHILD,
647                       (PVOID)&ulNumObj,
648         (PRCSTRUCT) &rcStruct) != SIM_RC_OK) {
649         IpsMessage(PSZMSGQUERYOBJFAIL1, RCStruct.ulRC);   /* Display Err Msg */
650         DosExit(EXIT_THREAD, RCStruct.ulRC);
651     }
652
653     if (ulNumObj == 0) {                    /* No objects returned from scan */
654         IpsMessage(PSZMSGNOSCANOBJ, RCStruct.ulRC);       /* Display Err Msg */
655         DosExit(EXIT_THREAD, RCStruct.ulRC);
656     }
657
658     /*************************************************************************/
659     /* Retrieve list of objects (pages) created during scan                  */
660     /*************************************************************************/
661
662     if ( SimWsQueryObj((SIMWSOBJ)hWsObj,
663                        SIMUSER_NONE,
664                        SIMWS_LIST_CHILD,
665                        (PVOID)&pScanObj,
666         (PRCSTRUCT) &rcStruct) != SIM_RC_OK) {
667         IpsMessage(PSZMSGQUERYOBJFAIL2, RCStruct.ulRC);   /* Display Err Msg */
668         DosExit(EXIT_THREAD, RCStruct.ulRC);
669     }
670
671     /*************************************************************************/
672     /* Initialize Document Attribute structure with the three basic standard */
673     /* attributes for the "INDEX_NOINDEX" index class.                       */
674     /*************************************************************************/
675
676     DosGetDateTime(&DateTime);                /* Date/Time structure       */
677     sprintf(szTimeStamp,                     /* String for Time Stamp     */
678             "%04d-%02d-%02d-%02d.%02d.%02d.%02d0000",
679             DateTime.year,
680             DateTime.month,
681             DateTime.day,
682             DateTime.hours,
683             DateTime.minutes,
684             DateTime.seconds,
685             DateTime.hundredths);
686
687     Docu[2].ulStruct = (Docu[1].ulStruct = (Docu[0].ulStruct = sizeof(Docu)));
688     Docu[0].usAttrId          = 40;                  /* AttrID for "Source" */
689     Docu[0].pszAttributeValue = "IP3SCAN";           /* This Program        */
690     Docu[1].usAttrId          = 41;                  /* AttrID for "UserID" */
691     Docu[1].pszAttributeValue = UserID;              /* This user           */
692     Docu[2].usAttrId          = 42;                  /* AttrID for "Source" */
693     Docu[2].pszAttributeValue = szTimeStamp;         /* Time Stamp          */
694
695     Docu[0].fAttrFlags = SIM_ATTR_READABLE |
```

```
                        SIM_ATTR_WRITEABLE;          /* Read/Write flags    */
    Docu[2].fAttrFlags = (Docu[2].fAttrFlags = Docu[1].fAttrFlags);
    Docu[0].usAttrType = SIM_ATTR_VSTRING;           /* Attribute data type */
    Docu[1].usAttrType = SIM_ATTR_VSTRING;           /* Attribute data type */
    Docu[2].usAttrType = SIM_ATTR_TIMESTAMP;         /* Attribute data type */

    /*************************************************************************/
    /* Call SimLibCreateItem to create a document ItemID and basic attrs.    */
    /*************************************************************************/

    if (SimLibCreateItem ( hSession,              /* Session Handle          */
                        SIM_DOCUMENT,            /* Item is a document      */
                        SIM_INDEX_NOINDEX,       /* Don't know index class  */
                        3,                       /* Number of attributes    */
                        (PATTRLISTSTRUCT) &Docu, /* ptr to attribute struct.*/
                        SIM_ACC_SYSTEM_CHOOSE,   /* system-assign privileges*/
                        (PASYNCCTLSTRUCT) NULL,  /* Synchronous operation   */
                        &RCStruct )              /* Return code structure   */
        == SIM_RC_OK) {
        strncpy( DocItemID,                       /* Save ItemID allocated   */
                 (PSZ)RCStruct.ulParam1, DOC_ID_SIZE);
        strcpy (msgText, DocItemID);
        IpsMessage(strcat(msgText, PSZMSGCREATEITEMOK), RCStruct.ulRC );
        SimLibFree( hSession,                  /* Folder Mgr session handle  */
                    (PVOID) RCStruct.ulParam1, /* Ptr to memory block to free*/
                    (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure   */
    }
    else {
        IpsMessage(PSZMSGCREATEITEMFAIL, RCStruct.ulRC); /* Display Error Msg*/
        DosExit(EXIT_THREAD, RCStruct.ulRC);
    }

    /*************************************************************************/
    /* Add each WS object as a new part to the document item created.        */
    /* Note that basic scan returns document as a single multi-page object.  */
    /*************************************************************************/

    memset (&Obj, '\0', sizeof(OBJ));          /* Initialize Object Handle   */
    Obj.ulStruct  = sizeof(Obj);
    Obj.ulPart    = 0;                        /* Let system create part #   */
    Obj.sVersion  = (SHORT)NULL;              /* Not yet supported          */
    strcpy(Obj.szItemID, DocItemID);           /* From SimLibCreateItem...   */

    for ( count = 0 ; count < ulNumObj; count ++ ) {
        Obj.ulPart    = 0;                    /* Let system create part #   */
        /*********************************************************************/
        /* Create a new library object to store each part returned from scan */
        /*********************************************************************/
        if (SimLibCreateObject( hSession,
                                (HOBJ) &Obj,
                                SIM_CC_MODCA_IS2,
                                (PSMS)NULL,
                                SIM_PRI_NORMAL,
                                SIM_CLOSE,
                                0,
                                (LONG) 0,
                                SIM_BASE,
                                (PVOID)NULL,
                                (PASYNCCTLSTRUCT)NULL,
                                (PRCSTRUCT) &rcStruct ) == SIM_RC_OK ) {
            Store = *((HOBJ)rcStruct.ulParam1);
            DataSrc.hSession     = hSession;
            DataSrc.Obj          = Store;
            DataSrc.ulAccess     = SIM_ACCESS_READ_WRITE;
```

```
760 |   |   |    DataSrc.ulPriority   = SIM_PRI_BACKGROUND;
761 |   |   |
762 |   |   |    SimLibFree(hSession,
763 |   |   |    (PVOID) rcStruct.ulParam1,
764 |   |   |    (PRCSTRUCT) &rcStruct );
765 |   |   └}
766 |   |   ┌else {
767 |   |   |    IpsMessage(PSZMSGCREATEOBJFAIL, RCStruct.ulRC);     /* Error msg */
768 |   |   |    DosExit(EXIT_THREAD, RCStruct.ulRC);
769 |   |   └}
770 |   |    /*********************************************************************/
771 |   |    /* Copy Working Set Object contents to newly created Library object  */
772 |   |    /*********************************************************************/
773 |   |    if ( SimWsStoreObj((SIMWSOBJ)pScanObj[count],
774 |   |                        SIM_CC_MODCA_IS2,
775 |   |                        FALSE,
776 |   |                        SIMSRC_SERVER_OBJ,
777 |   |                        (PVOID)&DataSrc,
778 |   |                        (PSIMSTORE) NULL,
779 |   |   ┌                    (PRCSTRUCT) &rcStruct ) != SIM_RC_OK) {
780 |   |   |    IpsMessage(PSZMSGSTOREOBJFAIL, RCStruct.ulRC);      /* Error Msg */
781 |   |   |    DosExit(EXIT_THREAD, RCStruct.ulRC);
782 |   |   └}
783 |   └}
784 |   SimOpsFreeMem((PVOID)&pScanObj,           /* Free memory for object list */
785 |                 (PRCSTRUCT) &rcStruct );
786 |   WinDestroyMsgQueue( hmqscan );              /* Destroy the message queue */
787 |   WinTerminate( habscan );                   /* Terminate PM environment  */
788 |   strcpy (msgText, DocItemID);
789 |   IpsMessage(strcat(msgText, PSZMSGCREATEITEMOK), RCStruct.ulRC );
790 |   DosExit(EXIT_THREAD, 0);
791 └}
792
```

```
793  /*BEGINPROLOGUE*********************************************************************/
794  /*                                                                                   */
795  /*  Function         : Cleanup                                                       */
796  /*                                                                                   */
797  /*  Description      : Clean up resources before program terminates.                 */
798  /*                                                                                   */
799  /*  Function type    :  VOID.                                                        */
800  /*                                                                                   */
801  /*  Input Parameters :  ( &RCSTRUCT )                                                */
802  /*                                                                                   */
803  /*  Synopsis         :  VOID Cleanup    (&RCSTRUCT)                                  */
804  /*                                                                                   */
805  /*ENDPROLOGUE***********************************************************************/
806
807  VOID Cleanup    ( PRCSTRUCT pRC ) {
808      if ( hWsObj ) {                          /* Working Set Object handle       */
809          SimWsDeleteObj((SIMWSOBJ)hWsObj,  /* Handle to working set object    */
810               SIMPURGE_ALL,                  /* Delete entire working set       */
811               (PRCSTRUCT) &RCStruct );       /* Pointer to RC data structure    */
812      }
813      if ( hSession ) {                            /* Library Manager Session   */
814          SimLibLogoff( hSession,                    /* Folder Mgr session handle */
815                        (PASYNCCTLSTRUCT)NULL,     /* Request synch. processing */
816                        (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure  */
817      }
818      if ( hwnd ) {                                /* PM Window                 */
819          WinDestroyWindow( hwnd );                  /* Destroy PM Window         */
820      }
821      if ( hmq ) {                                 /* PM Message Queue          */
822          WinDestroyMsgQueue( hmq );                 /* Destroy the message queue */
823      }
824      if ( hab ) {                                 /* PM                        */
825          WinTerminate( hab );                     /* Terminate PM environment  */
826      }
827      return;
828  }
829
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/*  Function         : IpsMessage                                                   */
/*                                                                                  */
/*  Description      : Service function to handle application messages.             */
/*                                                                                  */
/*  Function type    :  VOID.                                                       */
/*                                                                                  */
/*  Input Parameters :  (PSZ) Message Text, (ULONG) Error Code                      */
/*                                                                                  */
/*  Output Parameters:  None.                                                       */
/*                                                                                  */
/*  Synopsis         :  VOID IpsMessage (PSZ, ULONG)                                */
/*                                                                                  */
/*  Return values    :  None.                                                       */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

VOID IpsMessage ( PSZ pszMsgText, ULONG ulRC ) {

    char MsgText[80];     /* to format message text */
    sprintf(MsgText, "%s %d\n", pszMsgText, ulRC);

    /* for PM environment */

    WinMessageBox (HWND_DESKTOP,
                   HWND_DESKTOP,
                   MsgText,
                   "Information",
                   0,
                   MB_OK          |
                   MB_INFORMATION |
                   MB_APPLMODAL);

    /* for command line environment */

 /* printf(MsgText); */

    return;
}

/*************************************************************************************/
/* End of IP3SCAN.C                                                                  */
/*************************************************************************************/
```

B.4 IP3SCAN.DEF (11/04/93 14:44:30)

```
;*BEGINPROLOGUE*********************************************************************/
;*                                                                                  */
;* MODULE: IP3SCAN.DEF                                                             */
;*                                                                                  */
;* Description: IP3SCAN.DEF is used by program IP3SCAN.C.                          */
;*                                                                                  */
;* OS          : OS/2 2.1                                                          */
;* Compiler    : C Set/2                                                           */
;*                                                                                  */
;*ENDPROLOGUE***********************************************************************/

NAME IP3SCAN WINDOWAPI

DESCRIPTION 'IP/3 Building Blocks - IP3SCAN'

```

```
16  STUB    'OS2STUB.EXE'
17
18  CODE    MOVEABLE
19  DATA    MOVEABLE MULTIPLE
20
21  HEAPSIZE   64000        ; Must be non-zero to use Local memory manager
22  STACKSIZE  64000        ; Must be non-zero for SS == DS
23                          ; suggest 4k as minimum stacksize
24  ;*
25  ;* End of IP3SCAN.DEF
```

B.5 IP3SCAN.MAK (04/25/94 15:18:14)

```
1   #*****************************************************************************
2   #
3   # Make file for IP3SCAN.C                        ImagePlus VisualInfo
4   #
5   #*****************************************************************************
6   #
7   #  COPYRIGHT:
8   #  ----------
9   #  Copyright (C) International Business Machines Corp., 1993, 1994.
10  #
11  #
12  #  DISCLAIMER OF WARRANTIES:
13  #  -------------------------
14  #  The following [enclosed] code is sample code created by IBM
15  #  Corporation.  This sample code is not part of any standard IBM product
16  #  and is provided to you solely for the purpose of assisting you in the
17  #  development of your applications.  The code is provided "AS IS",
18  #  without warranty of any kind.  IBM shall not be liable for any damages
19  #  arising out of your use of the sample code, even if they have been
20  #  advised of the possibility of such damages.
21  #
22  #*****************************************************************************
23  #
24  # External Environment Variables Used
25  #
26  #   SB_ROOT       This environment variable defines where the
27  #                 VisualInfo directory is located.  It maybe set
28  #                 either in the config.sys file or from the
29  #                 command line. For example,
30  #
31  #                     SET SB_ROOT=D:\FRNV1RO
32  #
33  #   DEBUG         This environment variable defines whether a
34  #                 Debug or Non-Debug build is to be performed.
35  #                 It maybe set either in the config.sys file or
36  #                 from the command line(when switching back and
37  #                 forth is desired).  The variable may be set
38  #                 according to
39  #
40  #                     SET DEBUG=1  --> perform Debug build
41  #
42  #                     or,
43  #
44  #                     SET DEBUG=0  --> perform Non-Debug build
45  #
46  #*****************************************************************************
47  #
48  # Compiler Options Used
49  #
50  #  /C+          Compile only
```

```
#  /Fd-         Store internal work files in shared memory
#  /Gd+         Dynamically link the run-time library
#  /Ge+         Build an EXE file
#  /Ge-         Build a  DLL file
#  /Gh+         Generate code for profiling
#  /Gh-         Disable profiling
#  /Gm+         Link with multi-threaded version of library
#  /I           Include file location
#  /Kbcfogapexir  Give all diagnostics except preprocessor trace
#  /Lf-         Set all listing options off
#  /Mp          Use - optlink - linkage for functions
#  /Q+          Do not display logo
#  /Re          Generate executable code that can be used in an OS/2 runtime
#               environment
#  /Rn          Generate executable code that can be used as a subsystem with
#               no runtime environment
#  /Se          Allow all C Set ++ language extensions except migration
#  /Sm          Control Compiler interpretation of unsupported keywords
#  /Sn+         Allow use of DBCS
#  /Sp1         Align on single byte boundires (i.e pack)
#  /Sp4         Align on full word boundires (i.e don't pack)
#  /Ss          Allow double slash (//) for comments
#  /Ti+         Generate debugging information
#
#  /ALIGN       Set the alignment factor.  Must be power of 2.
#  /CO          Include symbolic debugger info
#  /DE          Prepare for debugging.
#  /EXEPACK     Compress byte pattern in *.exe
#  /MAP         Create *.map file.  Lisat public symbols.
#  /NOI         NOIGNORECASE           identifiers are case sensitive
#  /NOL         NOLOGO                 disable signon banner
#  /NOD         NODEFAULTLIBRARYSEARCH don't search LIB libraries
#  /PACKCODE    Pack neighboring code segments.
#  /PACKDATA    Pack neighboring data segments.
#
#*****************************************************************************

#├"
#] Generic/Common Macros                                                    ]
#Î┐
CCFLAGS  = /C+ /Gd+ /Ge+ /Gm+ /Kb /Mp /Q+ /Re /Se /Sn+ /Sp1 /Ss

INCLUDES = $(SB_ROOT)\INCLUDE;$(SB_ROOT)\INC

LLFLAGS  = /NOI /NOL

#├"
#] Set up defaults                                                          ]
#Î┐

!ifndef DEBUG
DEBUG=0
!endif

#├"
#] Debug/Non-Debug Macros                                                   ]
#Î┐
!if $(DEBUG)
CFLAGS   =  /Gh- /Lf- /Ti+ $(CCFLAGS) /I $(INCLUDES)
LFLAGS   = /CO /DE $(LLFLAGS)
BINDIR   = .
LIBDIR   = $(SB_ROOT)\LIB
SRCDIR   = .
OBJDIR   = .
SRCFIL   = IP3SCAN
!else
```

```
CFLAGS   = $(CCFLAGS) /I $(INCLUDES)
LFLAGS   = $(LLFLAGS)
BINDIR   = .
LIBDIR   = $(SB_ROOT)\LIB
SRCDIR   = .
OBJDIR   = .
SRCFIL   = IP3SCAN
!endif

COMPILE_REG = ICC $(CFLAGS) -Fo$(OBJDIR)\$@ $(SRCDIR)\$(@B).C

LINKLIBS = $(LIBDIR)\FRNOFI.LIB    \
           $(LIBDIR)\FRNOFO.LIB    \
           $(LIBDIR)\FIWSENV.LIB   \
           $(LIBDIR)\FIWSP.LIB     \
           $(LIBDIR)\FIWSWS.LIB    \
           $(LIBDIR)\FIWSD.LIB     \
           $(LIBDIR)\FIWSS         \
           OS2386.LIB

OFILES   = $(OBJDIR)\$(SRCFIL).OBJ

#├"
#] Dependencies                                                    ]
#Î┐
FOLDERMGR_DEP = {$(INCLUDES);}FRNP.H         {$(INCLUDES);}FRNPCAPI.H  \
                {$(INCLUDES);}FRNPERR.H      {$(INCLUDES);}FRNPFI.H    \
                {$(INCLUDES);}FRNPTYPE.H     {$(INCLUDES);}FRNPFI2.H   \
                {$(INCLUDES);}FRNPLIBC.H     {$(INCLUDES);}FRNPLCLI.H  \
                {$(INCLUDES);}FRNPFO.H

IS_DEP        = {$(INCLUDES);}FIWS.H         {$(INCLUDES);}FIWSDSP.H   \
                {$(INCLUDES);}FIWSWS.H       {$(INCLUDES);}FIWSENV.H

FRNOEXDI_DEP  = {$(INCLUDES);}$(SRCFIL).H     $(FOLDERMGR_DEP)          \
                 $(IS_DEP)

#├"
#] Main Target Rule                                                ]
#Î┐
ALL         : ERASE $(BINDIR)\$(SRCFIL).EXE

ERASE       :
    if exist ERR.LST erase ERR.LST

#├"
#] Rules                                                           ]
#Î┐
$(BINDIR)\$(SRCFIL).EXE   :  $(OFILES) $(SRCDIR)\$(SRCFIL).DEF $(SRCFIL).MAK
   LINK386 $(LFLAGS) $(OFILES),    \
           $(BINDIR)\$(SRCFIL).EXE, \
           $(BINDIR)\$(SRCFIL).MAP, \
           $(LINKLIBS),            \
           $(SRCDIR)\$(SRCFIL).DEF

$(OBJDIR)\$(SRCFIL).OBJ : $(SRCDIR)\$(SRCFIL).C $(FRNOEXDI_DEP)
   $(COMPILE_REG)
```

Appendix C. IP3IMPRT Source Code

C.1 Function index

DosGetDateTime	
called by	**ImportObject** (IP3IMPRT.C, page 106)

fclose	
called by	**main** (IP3IMPRT.C, page 102)

fopen	
called by	**main** (IP3IMPRT.C, page 102)

ImportObject (IP3IMPRT.C, page 106)	
calls	**DosGetDateTime** **IpsMessage** (IP3IMPRT.C, page 108) **SimLibCatalogObject** **SimLibCreateItem** **SimLibFree** **sprintf** **strcat** **strcpy** **strncpy**
called by	**main** (IP3IMPRT.C, page 102)

IpsMessage (IP3IMPRT.C, page 108)	
calls	**printf** **sprintf**
called by	**ImportObject** (IP3IMPRT.C, page 106) **main** (IP3IMPRT.C, page 102)

Logon (IP3IMPRT.C, page 105)	
calls	**SimLibLogon**
called by	**main** (IP3IMPRT.C, page 102)

main (IP3IMPRT.C, page 102)	
calls	**fclose** **fopen** **ImportObject** (IP3IMPRT.C, page 106) **IpsMessage** (IP3IMPRT.C, page 108) **Logon** (IP3IMPRT.C, page 105) **perror** **SimLibFree** **SimLibLogoff** **strcat** **strcpy** **strncpy** **strnicmp**

perror	
called by	**main** (IP3IMPRT.C, page 102)

printf	
called by	**IpsMessage** (IP3IMPRT.C, page 108)

SimLibCatalogObject	
called by	**ImportObject** (IP3IMPRT.C, page 106)

SimLibCreateItem	
called by	**ImportObject** (IP3IMPRT.C, page 106)

SimLibFree	
called by	**ImportObject** (IP3IMPRT.C, page 106) **main** (IP3IMPRT.C, page 102)

SimLibLogoff	
called by	**main** (IP3IMPRT.C, page 102)

SimLibLogon	
called by	**Logon** (IP3IMPRT.C, page 105)

sprintf	
called by	**ImportObject** (IP3IMPRT.C, page 106) **IpsMessage** (IP3IMPRT.C, page 108)

strncpy	
called by	**ImportObject** (IP3IMPRT.C, page 106) **main** (IP3IMPRT.C, page 102)

strcat	
called by	**ImportObject** (IP3IMPRT.C, page 106) **main** (IP3IMPRT.C, page 102)

strnicmp	
called by	**main** (IP3IMPRT.C, page 102)

strcpy	
called by	**ImportObject** (IP3IMPRT.C, page 106) **main** (IP3IMPRT.C, page 102)

C.2 IP3IMPRT.H (05/20/94 14:53:00)

```
1   /*****************************************************************************/
2   /*                                                                           */
3   /* MODULE: IP3IMPRT.H                                                        */
4   /*                                                                           */
5   /* Description  : Header file for module IP3IMPRT.C.                         */
6   /*                                                                           */
7   /*****************************************************************************/
8
9   /*****************************************************************************/
10  /* IP3IMPRT.C Function Prototypes                                            */
11  /*****************************************************************************/
12
13  ULONG    Logon              (PRCSTRUCT pRC,
14                               PSZ pszDBName,
15                               PSZ pszApplicationName,
16                               PSZ pszUserID,
17                               PSZ pszPassword) ;
18  ULONG    ImportObject        (PRCSTRUCT pRC) ;
19  VOID     IpsMessage          (PSZ   pszMsgText,
20                               ULONG ulRC) ;
21
22  /*****************************************************************************/
23  /* SimLibLogon Parameter Definitions                                         */
24  /*****************************************************************************/
25
26  #define PSZDBNAME          "LIBSRVR2"
27  #define PSZAPPLICATIONNAME "IPSAMP"
28  #define PSZUSERID          "FRNADMIN"
29  #define PSZPASSWORD        "PASSWORD"
30  #define PSZWBNAME          "TO BE INDEXED"    /* WorkBasket Name            */
31  #define PSZCLASSNAME       "NOINDEX"          /* Folder Manager Index Class */
32
33  /*****************************************/
34  /* SimLibLogon Parameter Maximum Lengths */
35  /*****************************************/
36
37  #define MAXDBNAME           18          /* Maximum size of pszDBName        */
38  #define MAXAPPLICATIONNAME  8           /* Maximum size of pszApplicationName*/
39  #define MAXUSERID           26          /* Maximum size of pszUserID        */
40  #define MAXPASSWORD         8           /* Maximum size of pszPassword      */
41
42  /*****************************************************************************/
43  /* SimOpsInitEnv and SimLibCatalogObject parameter default values            */
```

```
44  /*****************************************************************************/
45
46  #define PSZAPPLDESC         "Sample scenario to Export a document."
47  #define PSZWORKINGDIR       "D:\\FRNV1R0"
48  #define PSZEXITDIR          "D:\\FRNV1R0\\DLL"
49  #define PSZOIMANAGER        "FRNOFI"
50  #define PSZOMANAGER         "FRNOFO"
51
52  /*********************************************************/
53  /* SimOpsInitEnv and SimLibCatalogObject Maximum Lengths */
54  /*********************************************************/
55
56  #define MAXDIRNAME          255       /* Maximum size of pszWorkingDir,    */
57                                        /* pszExitDir, pszOIManager,         */
58                                        /* and pszOManager                   */
59  #define MAXFILENAME         255       /* Maximum size of pszFileName       */
60  #define MAXDESC             40        /* Maximum size of pszApplDesc       */
61                                        /* and pszDocDes                     */
62
63  /*****************************************/
64  /* Message Texts                         */
65  /*****************************************/
66
67  #define PSZMSGPARMERR       "Unrecognised parameter.  Parm number ="
68  #define PSZMSGFILEOPENFAIL  "Error opening file.  RC ="
69  #define PSZMSGLOGONFAIL     "Library Logon error.  RC ="
70  #define PSZMSGCREATEITEMFAIL "SimLibCreateItem error.  RC ="
71  #define PSZMSGCREATEITEMOK  " Document Item created.  RC ="
72  #define PSZMSGCATALOGOBJECTFAIL "SimLibCatalogObject error.  RC ="
73  #define PSZMSGCATALOGOBJECTOK "Object Catalogued successfully.  RC ="
74  #define PSZMSGIMPORTOBJECTFAIL "Error Importing Object.  RC ="
75  #define PSZMSGIMPORTOK      "Import request completed.  RC ="
76
77  /*****************************************************************************/
78  /* End of IP3IMPRT.H                                                         */
79  /*****************************************************************************/
```

C.3 IP3IMPRT.C (06/28/94 14:04:18)

ImportObject (IP3IMPRT.C, page 106)	
calls	**DosGetDateTime** **IpsMessage** (IP3IMPRT.C, page 108) **SimLibCatalogObject** **SimLibCreateItem** **SimLibFree** **sprintf** **strcat** **strcpy** **strncpy**
called by	**main** (IP3IMPRT.C, page 102)

IpsMessage (IP3IMPRT.C, page 108)	
calls	**printf** **sprintf**
called by	**ImportObject** (IP3IMPRT.C, page 106) **main** (IP3IMPRT.C, page 102)

Logon (IP3IMPRT.C, page 105)	
calls	**SimLibLogon**
called by	**main** (IP3IMPRT.C, page 102)

main (IP3IMPRT.C, page 102)	
calls	**fclose** **fopen** **ImportObject** (IP3IMPRT.C, page 106) **IpsMessage** (IP3IMPRT.C, page 108) **Logon** (IP3IMPRT.C, page 105) **perror** **SimLibFree** **SimLibLogoff** **strcat** **strcpy** **strncpy** **strnicmp**

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/* MODULE: IP3IMPRT.C                                                               */
/*                                                                                  */
/* DISCLAIMER OF WARRANTIES:                                                        */
/* -------------------------                                                        */
/* The following [enclosed] code is sample code created by IBM                      */
/* Corporation.  This sample code is not part of any standard IBM product           */
/* and is provided to you solely for the purpose of assisting you in the            */
/* development of your applications.  The code is provided "AS IS",                 */
/* without warranty of any kind.  IBM shall not be liable for any damages           */
/* arising out of your use of the sample code, even if they have been               */
/* advised of the possibility of such damages.                                      */
/*                                                                                  */
/* Description : IP3IMPRT.c - Sample program to export a document part              */
/*                            (object) to a PC file.                                */
/*               Folder Manager APIs used are : SimLibLogon, SimLibFree,            */
/*               SimLibCreateItem, SimLibCatalogObject                              */
/*                                                                                  */
/* System      : OS/2 compatible PC                                                 */
/* OS          : OS/2 2.1                                                           */
/* Compiler    : IBM C/C++                                                          */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

/************************************************************************************/
/*  System Header Files                                                             */
/************************************************************************************/
#define INCL_DOSDATETIME                /* Date and time values                     */
#include <os2.h>                        /* Main OS/2 header files                   */
#include <stdio.h>                      /* Standard I/O header files                */
#include <stdlib.h>                     /* Standard Library header files            */
#include <errno.h>                      /* Standard Error Handling                  */
#include <string.h>                     /* String manipulation                      */

/************************************************************************************/
/*  ImagePlus - Folder Manager header files                                         */
/************************************************************************************/
#define   FRN_INCL_FI                   /* Include frnpfi.h & frnplcli.h            */
#define   FRN_INCL_FO                   /* Include frnpfo.h                         */
#include "frnp.h"                       /* Global types                             */
#include "frnpcapi.h"                   /* Structures and constants                 */

/************************************************************************************/
/*  IP3IMPRT Header File                                                            */
/************************************************************************************/
#include "IP3IMPRT.h"                   /* Application header file                  */

/************************************************************************************/
/*  Global Variables                                                                */
/************************************************************************************/
CHAR     msgText[80]             = "";  /* Work string for building msg text */

/************************************************************************************/
/*  ImagePlus APIs - Library Session data used with Folder Manager APIs             */
/************************************************************************************/
HSESSION hSession;                      /* Folder Manager session handle            */
RCSTRUCT RCStruct;                      /* RC data structure for API calls          */
ULONG    ulMemoryBlock;                 /* Generic pointer for SimLibFree           */
CHAR     DBName[MAXDBNAME + 1]              = ""; /* Library Server Name           */
CHAR     ApplicationName[MAXAPPLICATIONNAME + 1]= ""; /* Application Name           */
CHAR     UserID[MAXUSERID + 1]              = ""; /* User ID                       */
CHAR     Password[MAXPASSWORD + 1]          = ""; /* Password                      */
CHAR     ClassName[SIM_CLASS_NAME_LENGTH + 1] = ""; /* Class Name                  */
CHAR     FileName[MAXFILENAME + 1]          = ""; /* File Name to import           */
CHAR     WBName[OIM_WB_NAME_LENGTH + 1]     = ""; /* WorkBasket name               */
```

```
67   ITEMID   DocItemID;                  /* ItemID of created document       */
68   OBJ      hObj;                       /* Object handle                    */
69
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/*  Function         : main                                                         */
/*                                                                                  */
/*  Description      : main() function of sample program to import a PC file        */
/*                     into an ImagePlus Library                                    */
/*                                                                                  */
/*  Function type    :  INT.                                                        */
/*                                                                                  */
/*  Input Parameters :  /N= ApplicationName                                         */
/*                      /U= UserID                                                  */
/*                      /P= Password                                                */
/*                      /S= DBName                                                  */
/*                      /F= FileName                                                */
/*                      /I= ClassName (not used)                                    */
/*                      /A  (not used)                                              */
/*                      /W= WBName (not used)                                       */
/*                                                                                  */
/*  Output Parameters:  None.                                                       */
/*                                                                                  */
/*  Synopsis         :  INT main ( int argc, char *argv[] )                         */
/*                                                                                  */
/*  Return values    :  If all functions are successful, the return code            */
/*                      from Logoff() is returned.  Otherwise, the return           */
/*                      code from the failing function is returned.                 */
/*                                                                                  */
/*  Process Flow     :                                                              */
/*                                                                                  */
/*      1.  Establish a session with the Folder Manager                             */
/*      2.  Create an item in the Library                                           */
/*      3.  Catalog the file as this library item.                                  */
/*      4.  Logoff from the Folder Manager session.                                 */
/*      5.  Exit main() with a return code.                                         */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

INT main (int argc, char *argv[]) {

    ULONG ulretcode = 0;                    /* Work variable for return codes */
    USHORT i;
    FILE   *pFile;                          /* File to be imported            */

    /*********************************************************************************/
    /* Initialize Session Variables required for logon with default values.  */
    /*********************************************************************************/
    strcpy (UserID,          PSZUSERID);
    strcpy (Password,        PSZPASSWORD);
    strcpy (DBName,          PSZDBNAME);
    strcpy (ApplicationName, PSZAPPLICATIONNAME);
    strcpy (WBName,          PSZWBNAME);

    /*********************************************************************************/
    /* Get the command line parameters                                               */
    /*********************************************************************************/

   for (i=1; i < argc; i++) {
      if (strnicmp(argv[i],"/N=",3)==0) {
         strncpy( ApplicationName, argv[i] + 3, MAXAPPLICATIONNAME );
      }
      else if (strnicmp(argv[i],"/U=",3)==0) {
         strncpy( UserID,          argv[i] + 3, MAXUSERID         );
      }
      else if (strnicmp(argv[i],"/S=",3)==0) {
         strncpy( DBName,          argv[i] + 3, MAXDBNAME         );
```

```
134     }
135     else if (strnicmp(argv[i],"/P=",3)==0) {
136         strncpy( Password,        argv[i] + 3, MAXPASSWORD        );
137     }
138     else if (strnicmp(argv[i],"/W=",3)==0) {
139         strncpy( WBName,          argv[i] + 3, OIM_WB_NAME_LENGTH );
140     }
141     else if (strnicmp(argv[i],"/I=",3)==0) {
142         strncpy( ClassName,       argv[i] + 3, SIM_CLASS_NAME_LENGTH );
143     }
144     else if (strnicmp(argv[i],"/F=",3)==0) {
145         strncpy( FileName,        argv[i] + 3, MAXFILENAME        );
146     }
147     else {
148         strcpy (msgText, argv[i]);
149         IpsMessage(strcat(msgText, PSZMSGPARMERR), (ULONG) i );
150         return(99);                          /* Quit Immediately      */
151     }
152 }
153
154  /*************************************************************************/
155  /* Check that file to be imported exists                                 */
156  /*************************************************************************/
157
158 if ((pFile = fopen(FileName, "r")) == NULL) {
159     IpsMessage(PSZMSGFILEOPENFAIL, errno);          /* Display Error Msg */
160     perror("");
161     RCStruct.ulRC = SIM_RC_ERROR_WRITING_TO_FILE;
162     return (errno);
163 }
164 else {
165     fclose(pFile);
166 }
167
168  /*************************************************************************/
169  /*  Call Logon to establish a Folder Manager Session.                    */
170  /* If successful, call SimLibFree to free the memory for structure       */
171  /* USERLOGONINFOSTRUC returned from SimLibLogon.                         */
172  /*************************************************************************/
173  if ( Logon ( &RCStruct,                        /* API Return Code struct*/
174               DBName,                          /* Library Server name   */
175               ApplicationName,                 /* Application Name      */
176               UserID,                          /* User ID               */
177               Password)                        /* Password              */
178                == SIM_RC_OK) {
179
180     SimLibFree( hSession,                  /* Folder Mgr session handle  */
181                 (PVOID) ulMemoryBlock,     /* Ptr to memory block to free*/
182                 (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure   */
183 }
184 else {
185     IpsMessage(PSZMSGLOGONFAIL, RCStruct.ulRC);
186     return(RCStruct.ulRC);                      /* Quit Immediately      */
187 }
188
189  /*************************************************************************/
190  /* Call ImportObject to create a Library Item and catalog the file in it */
191  /*************************************************************************/
192
193 if ( ImportObject( &RCStruct) != SIM_RC_OK ) {
194     IpsMessage(PSZMSGIMPORTOBJECTFAIL, RCStruct.ulRC);
195 }
```

```
196 |
197 |    /*************************************************************************/
198 |    /* Logoff Folder Manager and return.                                     */
199 |    /*************************************************************************/
200 |
201 |    ulretcode = SimLibLogoff( hSession,         /* Folder Mgr session handle */
202 |                   (PASYNCCTLSTRUCT)NULL,       /* Request synch. processing */
203 |                   (PRCSTRUCT) &RCStruct );     /* Ptr to RC data structure  */
204 |    return ( ulretcode );
205 └}
206
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : Logon                                                       */
/*                                                                                 */
/*  Description      : This function is called from main() to logon to             */
/*                     ImagePlus Bid Folder Manager.                               */
/*                                                                                 */
/*  Function type    :  ULONG                                                      */
/*                                                                                 */
/*  Input Parameters :  pszDBName pszApplicationName pszUserID pszPassword         */
/*                                                                                 */
/*  Output Parameters:  hSession RCSTRUCT                                          */
/*                                                                                 */
/*  Synopsis         :  ULONG Logon ( pRC                                          */
/*                                    pszDBName,                                   */
/*                                    pszApplicationName,                          */
/*                                    pszUserID,                                   */
/*                                    pszPassword)                                 */
/*                                                                                 */
/*  Return values    :  ulRC return code from SimLibLogon call                     */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*      1.  Initialize SimLibLogon parameters.                                     */
/*      2.  Establish a session with the Folder Manager                            */
/*          by calling SimLibLogon.                                                */
/*      3.  Return to main() with Session Handle                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG Logon (PRCSTRUCT pRC, PSZ pszDBName,
             PSZ pszApplicationName, PSZ pszUserID, PSZ pszPassword) {

    /*************************************************************************/
    /* Initialize SimLibLogon parameters.                                    */
    /*************************************************************************/

    RCStruct.ulRC      = SIM_RC_OK;

    /*************************************************************************/
    /* Call SimLibLogon to establish a Folder Manager session.               */
    /* If successful, save the Folder Manager session handle for subsequent  */
    /* Folder Manager API calls.                                             */
    /*************************************************************************/

    SimLibLogon( (PSZ) DBName,            /* Pointer to Database name        */
                 (PSZ) ApplicationName,   /* Pointer to Application Name     */
                 (PSZ) UserID,            /* Pointer to User Id              */
                 (PSZ) Password,          /* Pointer to Password for User Id */
                 (PSZ) NULL,              /* New Password                    */
                 (PSZ) NULL,              /* Logon without proxy             */
                 (PSZ) NULL,              /* Logon without proxy scope       */
                 SIM_SS_NORMAL,           /* Non-configuration session logon */
                 (PASYNCCTLSTRUCT) NULL,  /* Request synchronous processing  */
                 (PRCSTRUCT) &RCStruct ); /* Pointer to RC data structure    */

    if ( RCStruct.ulRC == SIM_RC_OK) {
        hSession = RCStruct.ulParam1;              /* Save Session Handle   */
                                                   /* Save ptr to USERLOGONINFOSTRUCT */
        ulMemoryBlock = RCStruct.ulParam2;         /* for use with SimLibFree */
    }
    return( RCStruct.ulRC );
}

```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : ImportObject                                                */
/*                                                                                 */
/*  Description      : This function is called from main() to perform the          */
/*                     functions necessary to export the first object in           */
/*                     a document.                                                 */
/*                                                                                 */
/*  Function type    :  ULONG.                                                     */
/*                                                                                 */
/*  Input Parameters :  *RCStruct                                                  */
/*                                                                                 */
/*  Output Parameters:  RCStruct                                                   */
/*                                                                                 */
/*  Synopsis         :  ULONG ExportObject( PRCSTRUCT pRC )                        */
/*                                                                                 */
/*  Return values    :  If all functions are successful, SIM_RC_OK is              */
/*                      returned.  Otherwise the return code from                  */
/*                      SimOpsInitEnv, SimLiboadObj, SimWsStoreObj                 */
/*                      or SimWsDeleteObj is returned.                             */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*      1.  Create a Library Item                                                  */
/*      2.  Catalog the file as an object in the Library Item.                     */
/*      3.  Return to main() with a return code.                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG ImportObject ( PRCSTRUCT pRC ) {

    ATTRLISTSTRUCT Docu [3];                  /* Document Attributes       */
    CHAR szTimeStamp[26];                     /* String for Time Stamp     */
    DATETIME   DateTime;                      /* For System Date/Time info */

    DosGetDateTime(&DateTime);                  /* Date/Time structure       */
    sprintf(szTimeStamp,                       /* String for Time Stamp     */
            "%04d-%02d-%02d-%02d.%02d.%02d.%02d0000",
            DateTime.year,
            DateTime.month,
            DateTime.day,
            DateTime.hours,
            DateTime.minutes,
            DateTime.seconds,
            DateTime.hundredths);

    RCStruct.ulStruct = sizeof(RCSTRUCT);

    /*************************************************************************/
    /* Initialize Document Attribute structure with the three basic standard */
    /* attributes for the "INDEX_NOINDEX" index class.                       */
    /*************************************************************************/

    Docu[2].ulStruct = (Docu[1].ulStruct = (Docu[0].ulStruct = sizeof(Docu)));
    Docu[0].usAttrId          = 40;                 /* AttrID for "Source" */
    Docu[0].pszAttributeValue = "IP3IMPRT";         /* This Program        */
    Docu[1].usAttrId          = 41;                 /* AttrID for "UserID" */
    Docu[1].pszAttributeValue = UserID;             /* This Program        */
    Docu[2].usAttrId          = 42;                 /* AttrID for "Source" */
    Docu[2].pszAttributeValue = szTimeStamp;        /* Time Stamp          */

    Docu[0].fAttrFlags = SIM_ATTR_READABLE |
                         SIM_ATTR_WRITEABLE;        /* Read/Write flags    */
    Docu[2].fAttrFlags = (Docu[2].fAttrFlags = Docu[1].fAttrFlags);
    Docu[0].usAttrType = SIM_ATTR_VSTRING;          /* Attribute data type */
    Docu[1].usAttrType = SIM_ATTR_VSTRING;          /* Attribute data type */
    Docu[2].usAttrType = SIM_ATTR_TIMESTAMP;        /* Attribute data type */
```

```
336 |
337 |    /*************************************************************************/
338 |    /* Call SimLibCreateItem to create a document ItemID and basic attrs.    */
339 |    /*************************************************************************/
340 |
341 |    SimLibCreateItem ( hSession,                 /* Session Handle          */
342 |                       SIM_DOCUMENT,            /* Item is a document      */
343 |                       SIM_INDEX_NOINDEX,       /* Don't know index class  */
344 |                       3,                       /* Number of attributes    */
345 |                       (PATTRLISTSTRUCT) &Docu, /* ptr to attribute struct.*/
346 |                       SIM_ACC_SYSTEM_CHOOSE,   /* system-assign privileges*/
347 |                       (PASYNCCTLSTRUCT) NULL,  /* Synchronous operation   */
348 |                       &RCStruct );             /* Return code structure   */
349 |    ┌if ( RCStruct.ulRC == SIM_RC_OK) {
350 |    │    strncpy( DocItemID,                    /* Save ItemID allocated   */
351 |    │             (PSZ)RCStruct.ulParam1, DOC_ID_SIZE);
352 |    │    strcpy (msgText, DocItemID);
353 |    │    IpsMessage(strcat(msgText, PSZMSGCREATEITEMOK), RCStruct.ulRC );
354 |    │    SimLibFree( hSession,                /* Folder Mgr session handle  */
355 |    │                (PVOID) RCStruct.ulParam1, /* Ptr to memory block to free*/
356 |    │                (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure   */
357 |    └}
358 |    ┌else {
359 |    │    IpsMessage(PSZMSGCREATEITEMFAIL, RCStruct.ulRC); /* Display Error Msg*/
360 |    │    return( RCStruct.ulRC );
361 |    └}
362 |
363 |    /*************************************************************************/
364 |    /* Catalog the specified file as an object in the item created.          */
365 |    /*************************************************************************/
366 |
367 |    hObj.ulStruct = sizeof(OBJ);
368 |    hObj.ulPart = 0;
369 |    hObj.sVersion = 0;
370 |    strcpy(hObj.szItemID, DocItemID);
371 |    strcpy(hObj.chRepType, "");
372 |    SimLibCatalogObject ( hSession,          /* Session handle              */
373 |                          &hObj,            /* Object handle structure     */
374 |                          SIM_CC_MODCA_IS2, /* Doc type is MO:DCA          */
375 |                          NULL,             /* pointer to SMS structure    */
376 |                          (PSZ) &FileName,  /* File to import              */
377 |                          SIM_PRI_DEFAULT,  /* Object priority index       */
378 |                          SIM_CLOSE,        /* Close object when complete  */
379 |                          0,                /* Version control flag        */
380 |                          0,                /* Part sequence               */
381 |                          SIM_BASE,         /* Affiliated type             */
382 |                          NULL,             /* ptr to affiliated data      */
383 |                          NULL,             /* Synchronous operation       */
384 |                          &RCStruct );
385 |    ┌if ( RCStruct.ulRC == SIM_RC_OK) {
386 |    │    IpsMessage(PSZMSGCATALOGOBJECTOK, RCStruct.ulRC );
387 |    │    SimLibFree( hSession,                /* Folder Mgr session handle  */
388 |    │                (PVOID) RCStruct.ulParam1, /* Ptr to memory block to free*/
389 |    │                (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure   */
390 |    └}
391 |    ┌else {
392 |    │    IpsMessage(PSZMSGCATALOGOBJECTFAIL, RCStruct.ulRC );
393 |    │    return( RCStruct.ulRC );
394 |    └}
395 |
396 |    return ( RCStruct.ulRC );
397 └}
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/*  Function         : IpsMessage                                                   */
/*                                                                                  */
/*  Description      : Service function to handle application messages.             */
/*                                                                                  */
/*  Function type    :  VOID.                                                       */
/*                                                                                  */
/*  Input Parameters :  (PSZ) Message Text, (ULONG) Error Code                      */
/*                                                                                  */
/*  Output Parameters:  None.                                                       */
/*                                                                                  */
/*  Synopsis         :  VOID IpsMessage (PSZ, ULONG)                                */
/*                                                                                  */
/*  Return values    :  None.                                                       */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

VOID IpsMessage ( PSZ pszMsgText, ULONG ulRC ) {

    char MsgText[80];     /* to format message text */
    sprintf(MsgText, "%s %d\n", pszMsgText, ulRC);

    /* for PM environment */

 /* WinMessageBox (HWND_DESKTOP,
                   HWND_DESKTOP,
                   MsgText,
                   "Information",
                   0,
                   MB_OK          |
                   MB_INFORMATION |
                   MB_APPLMODAL); */

    /* for command line environment */

    printf(MsgText);

    return;
}

/*************************************************************************************/
/* End of IP3IMPRT.C                                                                 */
/*************************************************************************************/
```

C.4 IP3IMPRT.DEF (10/25/93 10:42:00)

```
;*BEGINPROLOGUE*********************************************************************/
;*                                                                                  */
;* MODULE: IP3IMPRT.DEF                                                            */
;*                                                                                  */
;* Description: IP3IMPRT.DEF is used by program IP3IMPRT.C.                        */
;*                                                                                  */
;*ENDPROLOGUE***********************************************************************/

NAME IP3IMPRT WINDOWCOMPAT

DESCRIPTION 'IP/3 Building Blocks - IP3IMPRT'

STUB    'OS2STUB.EXE'

CODE    MOVEABLE
```

```
16  DATA    MOVEABLE MULTIPLE
17
18  HEAPSIZE   65536        ; Must be non-zero to use Local memory manager
19  STACKSIZE  65536        ; Must be non-zero for SS == DS
20                          ; suggest 4k as minimum stacksize
21  ;*
22  ;* End of IP3IMPRT.DEF
```

C.5 IP3IMPRT.MAK (04/14/94 15:06:08)

```
 1  #*****************************************************************************
 2  #
 3  # Make file for IP3IMPRT.C                        ImagePlus VisualInfo
 4  #
 5  #*****************************************************************************
 6  #
 7  #  COPYRIGHT:
 8  #  ----------
 9  #  Copyright (C) International Business Machines Corp., 1993, 1994.
10  #
11  #
12  #  DISCLAIMER OF WARRANTIES:
13  #  -------------------------
14  #  The following [enclosed] code is sample code created by IBM
15  #  Corporation.  This sample code is not part of any standard IBM product
16  #  and is provided to you solely for the purpose of assisting you in the
17  #  development of your applications.  The code is provided "AS IS",
18  #  without warranty of any kind.  IBM shall not be liable for any damages
19  #  arising out of your use of the sample code, even if they have been
20  #  advised of the possibility of such damages.
21  #
22  #*****************************************************************************
23  #
24  # External Environment Variables Used
25  #
26  #   SB_ROOT       This environment variable defines where the
27  #                 VisualInfo directory is located.  It maybe set
28  #                 either in the config.sys file or from the
29  #                 command line. For example,
30  #
31  #                     SET SB_ROOT=D:\FRNV1R0
32  #
33  #   DEBUG         This environment variable defines whether a
34  #                 Debug or Non-Debug build is to be performed.
35  #                 It maybe set either in the config.sys file or
36  #                 from the command line(when switching back and
37  #                 forth is desired).  The variable may be set
38  #                 according to
39  #
40  #                     SET DEBUG=1  --> perform Debug build
41  #
42  #                     or,
43  #
44  #                     SET DEBUG=0  --> perform Non-Debug build
45  #
46  #*****************************************************************************
47  #
48  # Compiler Options Used
49  #
50  #  /C+          Compile only
51  #  /Fd-         Store internal work files in shared memory
52  #  /Gd+         Dynamically link the run-time library
53  #  /Ge+         Build an EXE file
```

```
#  /Ge-         Build a  DLL file
#  /Gh+         Generate code for profiling
#  /Gh-         Disable profiling
#  /Gm+         Link with multi-threaded version of library
#  /I           Include file location
#  /Kbcfogapexir  Give all diagnostics except preprocessor trace
#  /Lf-         Set all listing options off
#  /Mp          Use - optlink - linkage for functions
#  /Q+          Do not display logo
#  /Re          Generate executable code that can be used in an OS/2 runtime
#               environment
#  /Rn          Generate executable code that can be used as a subsystem with
#               no runtime environment
#  /Se          Allow all C Set ++ language extensions except migration
#  /Sm          Control Compiler interpretation of unsupported keywords
#  /Sn+         Allow use of DBCS
#  /Sp1         Align on single byte boundires (i.e pack)
#  /Sp4         Align on full word boundires (i.e don't pack)
#  /Ss          Allow double slash (//) for comments
#  /Ti+         Generate debugging information
#
#  /ALIGN       Set the alignment factor.  Must be power of 2.
#  /CO          Include symbolic debugger info
#  /DE          Prepare for debugging.
#  /EXEPACK     Compress byte pattern in *.exe
#  /MAP         Create *.map file.  Lisat public symbols.
#  /NOI         NOIGNORECASE           identifiers are case sensitive
#  /NOL         NOLOGO                 disable signon banner
#  /NOD         NODEFAULTLIBRARYSEARCH don't search LIB libraries
#  /PACKCODE    Pack neighboring code segments.
#  /PACKDATA    Pack neighboring data segments.
#
#*******************************************************************************

#┌"
#] Generic/Common Macros                                                    ]
#Î┐
CCFLAGS  = /C+ /Gd+ /Ge+ /Gm+ /Kb /Mp /Q+ /Re /Se /Sn+ /Sp1 /Ss

INCLUDES = $(SB_ROOT)\INCLUDE;$(SB_ROOT)\INC

LLFLAGS  = /NOI /NOL

#┌"
#] Set up defaults                                                          ]
#Î┐

!ifndef DEBUG
DEBUG=0
!endif

#┌"
#] Debug/Non-Debug Macros                                                   ]
#Î┐
!if $(DEBUG)
CFLAGS   =  /Gh- /Lf- /Ti+ $(CCFLAGS) /I $(INCLUDES)
LFLAGS   = /CO /DE $(LLFLAGS)
BINDIR   = .
LIBDIR   = $(SB_ROOT)\LIB
SRCDIR   = .
OBJDIR   = .
SRCFIL   = IP3IMPRT
!else
CFLAGS   = $(CCFLAGS) /I $(INCLUDES)
LFLAGS   = $(LLFLAGS)
BINDIR   = .
```

```
LIBDIR   = $(SB_ROOT)\LIB
SRCDIR   = .
OBJDIR   = .
SRCFIL   = IP3IMPRT
!endif

COMPILE_REG = ICC $(CFLAGS) -Fo$(OBJDIR)\$@ $(SRCDIR)\$(@B).C

LINKLIBS = $(LIBDIR)\FRNOFI.LIB    \
           $(LIBDIR)\FRNOFO.LIB    \
           $(LIBDIR)\FIWSENV.LIB   \
           $(LIBDIR)\FIWSWS.LIB    \
           $(LIBDIR)\FIWSD.LIB     \
           OS2386.LIB

OFILES   = $(OBJDIR)\$(SRCFIL).OBJ

#├"
#] Dependencies                                                    ]
#Î┐

FOLDERMGR_DEP = {$(INCLUDES);}FRNP.H         {$(INCLUDES);}FRNPCAPI.H  \
                {$(INCLUDES);}FRNPERR.H      {$(INCLUDES);}FRNPFI.H    \
                {$(INCLUDES);}FRNPTYPE.H     {$(INCLUDES);}FRNPFI2.H   \
                {$(INCLUDES);}FRNPLIBC.H     {$(INCLUDES);}FRNPLCLI.H  \
                {$(INCLUDES);}FRNPFO.H

IS_DEP        = {$(INCLUDES);}FIWS.H         {$(INCLUDES);}FIWSDSP.H   \
                {$(INCLUDES);}FIWSWS.H       {$(INCLUDES);}FIWSENV.H

FRNOEXDI_DEP  = {$(INCLUDES);}$(SRCFIL).H     $(FOLDERMGR_DEP)          \
                 $(IS_DEP)

#├"
#] Main Target Rule                                                ]
#Î┐
ALL         : ERASE $(BINDIR)\$(SRCFIL).EXE

ERASE       :
    if exist ERR.LST erase ERR.LST

#├"
#] Rules                                                           ]
#Î┐
$(BINDIR)\$(SRCFIL).EXE   :  $(OFILES) $(SRCDIR)\$(SRCFIL).DEF $(SRCFIL).MAK
   LINK386 $(LFLAGS) $(OFILES),    \
           $(BINDIR)\$(SRCFIL).EXE, \
           $(BINDIR)\$(SRCFIL).MAP, \
           $(LINKLIBS),            \
           $(SRCDIR)\$(SRCFIL).DEF

$(OBJDIR)\$(SRCFIL).OBJ : $(SRCDIR)\$(SRCFIL).C $(FRNOEXDI_DEP)
   $(COMPILE_REG)
```

Appendix D. IP3DISP Source Code

D.1 Function index

Cleanup (IP3DISP.C, page 131)	
calls	**SimLibLogoff** **SimWsDeleteObj** **WinDestroyMsgQueue** **WinDestroyWindow** **WinTerminate**
called by	**main** (IP3DISP.C, page 120)

DisplayObject (IP3DISP.C, page 128)	
calls	**IpsMessage** (IP3DISP.C, page 132) **SimDspCreateWin** **SimDspSetWin** **SimOpsInitEnv** **SimWsLoadObj** **strcat** **strcpy**
called by	**main** (IP3DISP.C, page 120)

GetDocumentTOC (IP3DISP.C, page 126)	
calls	**IpsMessage** (IP3DISP.C, page 132) **SimLibGetItemAffiliatedTOC** **strcat**
called by	**main** (IP3DISP.C, page 120)

IP3DispWinProc1 (IP3DISP.C, page 124)	
calls	**WinBeginPaint** **WinDefWindowProc** **WinEndPaint** **WinFillRect**

IpsMessage (IP3DISP.C, page 132)	
calls	**sprintf** **WinMessageBox**
called by	**DisplayObject** (IP3DISP.C, page 128) **GetDocumentTOC** (IP3DISP.C, page 126) **main** (IP3DISP.C, page 120)

Logon (IP3DISP.C, page 125)	
calls	**SimLibLogon**
called by	**main** (IP3DISP.C, page 120)

main (IP3DISP.C, page 120)	
calls	**Cleanup** (IP3DISP.C, page 131) **DisplayObject** (IP3DISP.C, page 128) **GetDocumentTOC** (IP3DISP.C, page 126) **IpsMessage** (IP3DISP.C, page 132) **Logon** (IP3DISP.C, page 125) **QueryDocument** (IP3DISP.C, page 127) **SimLibFree** **strcat** **strcpy** **strlen** **strncpy** **strnicmp** **WinCreateMsgQueue** **WinCreateStdWindow** **WinDispatchMsg** **WinGetMsg** **WinInitialize** **WinRegisterClass**

QueryDocument (IP3DISP.C, page 127)	
calls	**SimLibQueryObject**
called by	**main** (IP3DISP.C, page 120)

SimDspCreateWin	
called by	**DisplayObject** (IP3DISP.C, page 128)

SimDspSetWin	
called by	**DisplayObject** (IP3DISP.C, page 128)

SimLibFree	
called by	**main** (IP3DISP.C, page 120)

SimLibGetItemAffiliatedTOC	
called by	**GetDocumentTOC** (IP3DISP.C, page 126)

SimLibLogoff	
called by	**Cleanup** (IP3DISP.C, page 131)

SimLibLogon	
called by	**Logon** (IP3DISP.C, page 125)

SimLibQueryObject	
called by	**QueryDocument** (IP3DISP.C, page 127)

SimOpsInitEnv	
called by	**DisplayObject** (IP3DISP.C, page 128)

SimWsDeleteObj	
called by	**Cleanup** (IP3DISP.C, page 131)

SimWsLoadObj	
called by	**DisplayObject** (IP3DISP.C, page 128)

sprintf	
called by	**IpsMessage** (IP3DISP.C, page 132)

strcat	
called by	**DisplayObject** (IP3DISP.C, page 128) **GetDocumentTOC** (IP3DISP.C, page 126) **main** (IP3DISP.C, page 120)

strcpy	
called by	**DisplayObject** (IP3DISP.C, page 128) **main** (IP3DISP.C, page 120)

strlen	
called by	**main** (IP3DISP.C, page 120)

strncpy	
called by	**main** (IP3DISP.C, page 120)

strnicmp	
called by	**main** (IP3DISP.C, page 120)

WinBeginPaint	
called by	**IP3DispWinProc1** (IP3DISP.C, page 124)

WinCreateMsgQueue	
called by	**main** (IP3DISP.C, page 120)

WinCreateStdWindow	
called by	**main** (IP3DISP.C, page 120)

WinDefWindowProc	
called by	**IP3DispWinProc1** (IP3DISP.C, page 124)

WinDestroyMsgQueue	
called by	**Cleanup** (IP3DISP.C, page 131)

WinDestroyWindow	
called by	**Cleanup** (IP3DISP.C, page 131)

WinDispatchMsg	
called by	**main** (IP3DISP.C, page 120)

WinEndPaint	
called by	**IP3DispWinProc1** (IP3DISP.C, page 124)

WinFillRect	
called by	**IP3DispWinProc1** (IP3DISP.C, page 124)

WinGetMsg	
called by	**main** (IP3DISP.C, page 120)

WinInitialize	
called by	**main** (IP3DISP.C, page 120)

WinMessageBox	
called by	**IpsMessage** (IP3DISP.C, page 132)

WinRegisterClass	
called by	**main** (IP3DISP.C, page 120)

WinTerminate	
called by	**Cleanup** (IP3DISP.C, page 131)

D.2 IP3DISP.H (05/20/94 14:54:58)

```
/*****************************************************************************/
/*                                                                           */
/* MODULE: IP3DISP.H                                                         */
/*                                                                           */
/* Description  : Header file for module IP3DISP.C.                          */
/*                                                                           */
/*****************************************************************************/

/*****************************************************************************/
/* IP3DISP.C Function Prototypes                                             */
/*****************************************************************************/

ULONG    Logon              (PRCSTRUCT pRC,
                             PSZ pszDBName,
                             PSZ pszApplicationName,
                             PSZ pszUserID,
                             PSZ pszPassword) ;
ULONG    GetDocumentTOC      (HSESSION hSession,
                             PRCSTRUCT pRC) ;
ULONG    QueryDocument       (HSESSION hSession,
                             PRCSTRUCT pRC) ;
ULONG    DisplayObject       (PRCSTRUCT pRC) ;
VOID     Cleanup             (PRCSTRUCT pRC) ;
VOID     IpsMessage          (PSZ   pszMsgText,
                             ULONG ulRC) ;

/*****************************************************************************/
/* SimLibLogon Parameter Definitions                                         */
/*****************************************************************************/

#define PSZDBNAME          "LIBSRVR2"
#define PSZAPPLICATIONNAME "SAMPLE"
#define PSZUSERID          "FRNADMIN"
#define PSZPASSWORD        "PASSWORD"

/*****************************************/
/* SimLibLogon Parameter Maximum Lengths */
/*****************************************/

#define MAXDBNAME          18          /* Maximum size of pszDBName         */
#define MAXAPPLICATIONNAME  8          /* Maximum size of pszApplicationName*/
#define MAXUSERID          26          /* Maximum size of pszUserID         */
#define MAXPASSWORD        8           /* Maximum size of pszPassword       */

/*****************************************************************************/
/* SimOpsInitEnv Parameter Definitions                                       */
/*****************************************************************************/

#define PSZAPPLDESC        "Sample scenario to print a document."
#define PSZWORKINGDIR      "D:\\FRNV1RO"
#define PSZEXITDIR         "D:\\FRNV1RO\\DLL"
#define PSZOIMANAGER       "FRNOFI"
#define PSZOMANAGER        "FRNOFO"

```

```
55  /***********************************************/
56  /* SimOpsInitEnv and SimWsLoadObj Maximum Lengths */
57  /***********************************************/
58
59  #define MAXDIRNAME          255        /* Maximum size of pszWorkingDir,    */
60                                         /* pszExitDir, pszOIManager,         */
61                                         /* and pszOManager                   */
62  #define MAXDESC             40         /* Maximum size of pszApplDesc       */
63                                         /* and pszDocDes                     */
64
65  /*****************************************/
66  /* Message Texts                         */
67  /*****************************************/
68
69  #define PSZMSGPARMERR       "Parameter Unrecognized, Parm # "
70  #define PSZMSGITEMERR       "ItemID should be 16 bytes long, is "
71  #define PSZMSGQRYDOCFAIL    "Query Document error.  RC ="
72  #define PSZMSGQRYTOCFAIL    "Query Document TOC error.  RC ="
73  #define PSZMSGLOGONFAIL     "Library Logon error.  RC ="
74  #define PSZMSGNODOCPART     "There are no parts for document "
75  #define PSZMSGOPSINITFAIL   "SimOpsInitEnv error.  RC ="
76  #define PSZMSGWSLOADOBJFAIL "SimWsLoadObj error.  RC ="
77  #define PSZMSGDISPFAIL      "Display Object error.  RC ="
78  #define PSZMSGDISPOBJECTFAIL"SimDspCreateWin error.  RC ="
79  #define PSZMSGDISPSETDILFAIL"SimDspSetWin (set DIL) error.  RC ="
80  #define PSZMSGDISPSHOWFAIL  "SimDspSetWin (show window) error.  RC ="
81
82  #define PSZTITLEBARTEXT     "Display of ItemID "
83
84  /*****************************************************************************/
85  /* PM Stuff                                                                  */
86  /*****************************************************************************/
87
88  #define IDFRAME 100
89  MRESULT EXPENTRY IP3DispWinProc1(HWND hwnd, ULONG msg, MPARAM mp1, MPARAM mp2);
90
91  /*****************************************************************************/
92  /* End of IP3DISP.H                                                          */
93  /*****************************************************************************/
```

D.3 IP3DISP.C (01/10/95 14:28:14)

Cleanup (IP3DISP.C, page 131)	
calls	**SimLibLogoff** **SimWsDeleteObj** **WinDestroyMsgQueue** **WinDestroyWindow** **WinTerminate**
called by	**main** (IP3DISP.C, page 120)

DisplayObject (IP3DISP.C, page 128)	
calls	**IpsMessage** (IP3DISP.C, page 132) **SimDspCreateWin** **SimDspSetWin** **SimOpsInitEnv** **SimWsLoadObj** **strcat** **strcpy**
called by	**main** (IP3DISP.C, page 120)

GetDocumentTOC (IP3DISP.C, page 126)	
calls	**IpsMessage** (IP3DISP.C, page 132) **SimLibGetItemAffiliatedTOC** **strcat**
called by	**main** (IP3DISP.C, page 120)

IP3DispWinProc1 (IP3DISP.C, page 124)	
calls	**WinBeginPaint** **WinDefWindowProc** **WinEndPaint** **WinFillRect**

IpsMessage (IP3DISP.C, page 132)	
calls	**sprintf** **WinMessageBox**
called by	**DisplayObject** (IP3DISP.C, page 128) **GetDocumentTOC** (IP3DISP.C, page 126) **main** (IP3DISP.C, page 120)

Logon (IP3DISP.C, page 125)	
calls	**SimLibLogon**
called by	**main** (IP3DISP.C, page 120)

main (IP3DISP.C, page 120)	
calls	**Cleanup** (IP3DISP.C, page 131) **DisplayObject** (IP3DISP.C, page 128) **GetDocumentTOC** (IP3DISP.C, page 126) **IpsMessage** (IP3DISP.C, page 132) **Logon** (IP3DISP.C, page 125) **QueryDocument** (IP3DISP.C, page 127) **SimLibFree** **strcat** **strcpy** **strlen** **strncpy** **strnicmp** **WinCreateMsgQueue** **WinCreateStdWindow** **WinDispatchMsg** **WinGetMsg** **WinInitialize** **WinRegisterClass**

QueryDocument (IP3DISP.C, page 127)	
calls	**SimLibQueryObject**
called by	**main** (IP3DISP.C, page 120)

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/* MODULE: IP3DISP.C                                                                */
/*                                                                                  */
/* DISCLAIMER OF WARRANTIES:                                                        */
/* -------------------------                                                        */
/* The following [enclosed] code is sample code created by IBM                      */
/* Corporation.  This sample code is not part of any standard IBM product           */
/* and is provided to you solely for the purpose of assisting you in the            */
/* development of your applications.  The code is provided "AS IS",                 */
/* without warranty of any kind.  IBM shall not be liable for any damages           */
/* arising out of your use of the sample code, even if they have been               */
/* advised of the possibility of such damages.                                      */
/*                                                                                  */
/* Description : IP3DISP.C - Sample application code to display a document.         */
/*               Folder Manager APIs used are : SimLibLogon, SimLibFree,            */
/*               SimLibGetItemAffiliatedTOC, SimLibQueryObject and                  */
/*               SimLibLogoff.                                                      */
/*               Image Services APIs used are: SimOpsInitEnv,                       */
/*               SimWsLoadObj, SimDspCreateWin and SimWsDeleteObj.                  */
/*                                                                                  */
/* System      : OS/2 compatible PC                                                 */
/* OS          : OS/2 2.1                                                           */
/* Compiler    : IBM C/C++                                                          */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

/************************************************************************************/
/*  System Header Files                                                             */
/************************************************************************************/
#define INCL_WIN
#include <os2.h>                        /* Main OS/2 header files           */
#include <stdio.h>                      /* Standard I/O header files        */
#include <stdlib.h>                     /* Standard Library header files    */
#include <string.h>                     /* String manipulation              */

/************************************************************************************/
/*  ImagePlus - Folder Manager header files                                         */
/************************************************************************************/
#define   FRN_INCL_FI                   /* Include frnpfi.h & frnplcli.h    */
#define   FRN_INCL_FO                   /* Include frnpfo.h                 */
#include "frnp.h"                       /* Global types                     */
#include "frnpcapi.h"                   /* Structures and constants         */

/************************************************************************************/
/*  ImagePlus - Image Services Header Files                                         */
/************************************************************************************/
#define   FIWS_SERVER                   /* Required to read from a server   */
#include "fiws.h"                       /* Base Image Services              */
#include "fiwsenv.h"                    /* Environment prototypes           */
#include "fiwsws.h"                     /* Working Set prototypes           */
#include "fiwsdsp.h"                    /* Display Prototypes               */

/************************************************************************************/
/*  IP3DISP Header File                                                             */
/************************************************************************************/
#include "IP3DISP.h"                    /* Application header file          */

/************************************************************************************/
/*  Global Variables                                                                */
/************************************************************************************/
/*  OS/2 and PM APIs                                                                */
/************************************************************************************/
HAB       hab   = 0;                    /* Anchor block handle              */
HMQ       hmq   = 0;                    /* Message Queue handle             */
HWND      hwnd  = 0;                    /* Main Window handle               */
```

```
67  HWND      hwndc = 0;                    /* handle to client area            */
68  QMSG      qmsg;                         /* Message Queue                    */
69  ULONG     ulCreateFlags;                /* Window frame controls            */
70
71  /*******************************************************************************/
72  /*  ImagePlus APIs - Library Session data used with Folder Manager APIs      */
73  /*******************************************************************************/
74  HSESSION hSession = 0;                  /* Folder Manager session handle    */
75  RCSTRUCT RCStruct;                      /* RC data structure for API calls  */
76  ULONG    ulMemoryBlock;                 /* Generic pointer for SimLibFree   */
77  CHAR     DBName[MAXDBNAME + 1] = "";               /* Library Server Name   */
78  CHAR     ApplicationName[MAXAPPLICATIONNAME + 1] = ""; /* Application Name  */
79  CHAR     UserID[MAXUSERID + 1] = "";               /* User ID               */
80  CHAR     Password[MAXPASSWORD + 1] = "";           /* Password              */
81  ITEMID   DocItemID = "";                /* Document to display              */
82  BOOL     fPartExists;                   /* Does a part exist for the document*/
83  OBJ      Obj;                           /* Document part to display         */
84  ULONG    ulObjConCls;                   /* Document content class           */
85
86  /*******************************************************************************/
87  /*  ImagePlus APIs - Image Services Data - used with Image Services APIs     */
88  /*******************************************************************************/
89  SIMWSOBJ hWsObj = 0;                    /* Working set object handle        */
90  SIMWIN    hWin  = 0;                    /* Display window handle            */
91
92  /*******************************************************************************/
93  /*  Define Parameters for SimWsLoadObj                                       */
94  /*******************************************************************************/
95
96  SIMCONTROL_BASE    pControl;            /* Object load control information  */
97  SIMSRCSERVEROBJ    pDataSrc;            /* Source from which to load data   */
98  SIMLOAD            pLoadID;             /* Pointer to load handle           */
99
100 /*******************************************************************************/
101 /*  Define Parameters for SimDspCreateWin and SimDspSetWin                   */
102 /*******************************************************************************/
103
104 CHAR          szTitle[40];              /* Display window title             */
105 ULONG         ulCreateFlags;            /* Window frame controls            */
106 SIMPOSSIZE    PosSize;                  /* Initial window position, size    */
107 BOOL          fSimDsp = TRUE;           /* Window Visibility Flag           */
108 PSZ           pszDilFmt =               /* define Dynamic Information Line  */
109               "Page %2PageInDoc. of %2NumPageInDoc.";
110
```

```
111  /*BEGINPROLOGUE*********************************************************/
112  /*                                                                      */
113  /*  Function         : main                                             */
114  /*                                                                      */
115  /*  Description      : main() function of sample program to display a doc.  */
116  /*                                                                      */
117  /*  Function type    :  INT.                                            */
118  /*                                                                      */
119  /*  Input Parameters :  /D= ItemID of document to display               */
120  /*                      /N= ApplicationName                             */
121  /*                      /U= UserID                                      */
122  /*                      /P= Password                                    */
123  /*                      /S= DBName (Library Server name)                */
124  /*                                                                      */
125  /*  Output Parameters:  None.                                           */
126  /*                                                                      */
127  /*  Synopsis         :  INT main ( int argc, char *argv[] )             */
128  /*                                                                      */
129  /*  Return values    :  If all functions are successful, the return code   */
130  /*                      from Logoff() is returned.  Otherwise, the return   */
131  /*                      code from the failing function is returned.     */
132  /*                                                                      */
133  /*  Process Flow     :                                                  */
134  /*      1.  Initialize Presentation Manager.                            */
135  /*      2.  Establish a session with the Folder Manager                 */
136  /*      3.  If base objects exist for the document:                     */
137  /*          objects exist for the document :                            */
138  /*          - Save the Obj structure of the first document part as      */
139  /*            the document that will be displayed.  Free the memory     */
140  /*            containing the array of PAFFTOCENTRYSTRUCT structures      */
141  /*          - Query the object to obtain the document content class     */
142  /*          - Free the memory containing the OBJINFOSTRUCT structure.   */
143  /*      4.  Create a PM Message Queue and a window.                     */
144  /*      5.  Display the first (or only) object in the document.         */
145  /*      6.  Process the message loop to wait for user to quit.          */
146  /*      7.  Free the working set.                                       */
147  /*      8.  Logoff from the Folder Manager session.                     */
148  /*      9.  Exit main() with a return code.                             */
149  /*                                                                      */
150  /*ENDPROLOGUE***********************************************************/
151
152  INT main (int argc, char *argv[]) {
153
154      ULONG  ulretcode   = 0;
155      USHORT i;
156      CHAR   msgText[80] = "";            /* Work string for building msg text */
157
158      /************************************************************************/
159      /* Initialize PM, and create Message Queue so we can use WinMessageBox.  */
160      /* PM is required for Image Services "Display" function to operate.      */
161      /************************************************************************/
162
163      hab = WinInitialize((USHORT) NULL);
164      hmq = WinCreateMsgQueue( hab, 0 );
165      if (hab == NULLHANDLE) {
166          return(99);                                 /* Quit Immediately      */
167      }
168
169      /************************************************************************/
170      /* Initialize Session Variables required for logon.                     */
171      /************************************************************************/
172      strcpy (DBName,           PSZDBNAME);
173      strcpy (ApplicationName,  PSZAPPLICATIONNAME);
174      strcpy (UserID,           PSZUSERID);
175      strcpy (Password,         PSZPASSWORD);
```

```
176
177     /*************************************************************************/
178     /* Check that Document ID has been passed as parameter.                  */
179     /*************************************************************************/
180
181     for (i=1; i < argc; i++) {
182         if (strnicmp(argv[i],"/D=",3)==0) {
183             strncpy( DocItemID, argv[i] + 3, DOC_ID_SIZE);
184         }
185         else if (strnicmp(argv[i],"/N=",3)==0) {
186             strncpy( ApplicationName, argv[i] + 3, MAXAPPLICATIONNAME );
187         }
188         else if (strnicmp(argv[i],"/U=",3)==0) {
189             strncpy( UserID,          argv[i] + 3, MAXUSERID          );
190         }
191         else if (strnicmp(argv[i],"/S=",3)==0) {
192             strncpy( DBName,          argv[i] + 3, MAXDBNAME          );
193         }
194         else if (strnicmp(argv[i],"/P=",3)==0) {
195             strncpy( Password,        argv[i] + 3, MAXPASSWORD        );
196         }
197         else {
198             IpsMessage(strcat(strcpy(msgText, argv[i]), PSZMSGPARMERR), i );
199             Cleanup( &RCStruct );
200             return(99);                              /* Quit Immediately     */
201         }
202     }
203     if (strlen(DocItemID) != DOC_ID_SIZE - 1) {
204         IpsMessage(PSZMSGITEMERR, strlen(DocItemID));
205         Cleanup( &RCStruct );
206         return(99);                                  /* Quit Immediately     */
207     }
208
209     /*************************************************************************/
210     /*  Call Logon to establish a Folder Manager Session.                    */
211     /* If successful, call SimLibFree to free the memory for structure       */
212     /* USERLOGONINFOSTRUC returned from SimLibLogon.                         */
213     /*************************************************************************/
214
215     ulretcode = Logon ( &RCStruct,                /* API Return Code struct*/
216                         DBName,                  /* Library Server name   */
217                         ApplicationName,         /* Application Name      */
218                         UserID,                  /* User ID               */
219                         Password);               /* Password              */
220
221     if (ulretcode == SIM_RC_OK) {
222
223         SimLibFree( hSession,                 /* Folder Mgr session handle  */
224                     (PVOID) ulMemoryBlock,    /* Ptr to memory block to free*/
225                     (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure   */
226     }
227     else {
228         IpsMessage(PSZMSGLOGONFAIL, ulretcode);
229         Cleanup( &RCStruct );
230         return(ulretcode);                           /* Quit Immediately     */
231     }
232
233     /*************************************************************************/
234     /*  Call GetDocumentTOC to find the document parts.                      */
235     /*  If successful, and base parts exist for the document, call           */
236     /*  FreeMemory to free the memory for an array of AFFTOCENTRYSTRUCT      */
237     /*  structures returned from SimLibGetItemAffiliatedTOC.                 */
```

```
238     /*************************************************************************/
239
240     ulretcode = GetDocumentTOC( hSession, &RCStruct );
241
242     if ( (ulretcode == SIM_RC_OK) && (fPartExists == TRUE) ) {
243         SimLibFree( hSession,                 /* Folder Mgr session handle  */
244                     (PVOID) ulMemoryBlock,    /* Ptr to memory block to free*/
245                     (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure   */
246     }
247     else {
248         IpsMessage(PSZMSGQRYTOCFAIL, ulretcode);
249         Cleanup( &RCStruct );
250         return(ulretcode);                    /* Quit Immediately      */
251     }
252
253     /*************************************************************************/
254     /*  Call QueryDocument to get the content class of the document part.    */
255     /* If successful, call FreeMemory to free the memory for structure       */
256     /* OBJINFOSTRUCT returned from SimLibQueryObject.                        */
257     /*************************************************************************/
258
259     ulretcode = QueryDocument( hSession, &RCStruct );
260
261     if (ulretcode == SIM_RC_OK) {
262         SimLibFree( hSession,                 /* Folder Mgr session handle  */
263                     (PVOID) ulMemoryBlock,    /* Ptr to memory block to free*/
264                     (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure   */
265     }
266     else {
267         IpsMessage(PSZMSGQRYDOCFAIL, ulretcode);
268         Cleanup( &RCStruct );
269         return(ulretcode);                    /* Quit Immediately      */
270     }
271
272     /*************************************************************************/
273     /* Create a parent window for IP3DISP.                                   */
274     /*************************************************************************/
275
276     WinRegisterClass (hab,
277                       "IP3DispMainWindow",
278                       (PFNWP) IP3DispWinProc1,
279                       CS_SIZEREDRAW,
280                       0);
281     ulCreateFlags = FCF_TITLEBAR      |
282                     FCF_SYSMENU       |
283                     FCF_SIZEBORDER    |
284                     FCF_MINMAX        |
285                     FCF_SHELLPOSITION |
286                     FCF_TASKLIST;
287     hwnd = WinCreateStdWindow (HWND_DESKTOP,
288                                WS_VISIBLE,
289                                &ulCreateFlags,
290                                "IP3DispMainWindow",
291                                "IP3Disp Main Window",
292                                0L,
293                                (HMODULE) NULL,
294                                IDFRAME,
295                                (PHWND) &hwndc);
296
297     /*************************************************************************/
298     /* Call DisplayObject to call Image Services to display the document.    */
299     /*************************************************************************/
300
301     ulretcode = DisplayObject( &RCStruct);
```

```
302
303     if (ulretcode != SIM_RC_OK) {
304         IpsMessage(PSZMSGDISPFAIL, ulretcode);
305     }
306
307     /*************************************************************************/
308     /* Drop into the PM message loop.                                        */
309     /*************************************************************************/
310     /*************************************************************************/
311     /*   Get and Dispatch messages from the application message queue until  */
312     /*   WinGetMsg returns FALSE, indicating a WM_QUIT message.              */
313     /*************************************************************************/
314
315     while (WinGetMsg (hab,
316                       &qmsg,
317                       (HWND) NULL,
318                       0,
319                       0 )) {
320         WinDispatchMsg (hab,
321                         &qmsg );
322     }
323
324     /*************************************************************************/
325     /*   Clean house and terminate the application                           */
326     /*************************************************************************/
327
328     Cleanup( &RCStruct );
329
330     return ( ulretcode );
331 }
332
```

```
333  /*BEGINPROLOGUE*********************************************************/
334  /*                                                                      */
335  /*  Function         : IP3DispWinProc1                                  */
336  /*                                                                      */
337  /*  Description      : Window proc for dummy window displayed after document */
338  /*                     is displayed.  This provides the means to terminate   */
339  /*                     IP3DISP after the Image Services Display is complete. */
340  /*                                                                      */
341  /*ENDPROLOGUE***********************************************************/
342
343  MRESULT EXPENTRY IP3DispWinProc1 (HWND hwnd, ULONG msg, MPARAM mp1, MPARAM mp2) {
344
345      HPS hps;
346      RECTL rectl;
347
348      switch ( msg ) {
349          case WM_CREATE:
350             break;
351
352          case WM_PAINT:
353              hps = WinBeginPaint(hwnd, 0L, (PRECTL) &rectl);
354              WinFillRect(hps, (PRECTL) &rectl, SYSCLR_WINDOW );
355              WinEndPaint(hps);
356              break;
357
358         case WM_COMMAND:
359              break;
360
361         default:
362             return(WinDefWindowProc (hwnd, msg, mp1, mp2));
363      }
364      return(FALSE);
365  }
366
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : Logon                                                       */
/*                                                                                 */
/*  Description      : This function is called from main() to logon to             */
/*                     ImagePlus Bid Folder Manager.                               */
/*                                                                                 */
/*  Function type    :  ULONG                                                      */
/*                                                                                 */
/*  Input Parameters :  pszDBName pszApplicationName pszUserID pszPassword         */
/*                                                                                 */
/*  Output Parameters:  hSession RCSTRUCT                                          */
/*                                                                                 */
/*  Synopsis         :  ULONG Logon ( pRC                                          */
/*                                    pszDBName,                                   */
/*                                    pszApplicationName,                          */
/*                                    pszUserID,                                   */
/*                                    pszPassword)                                 */
/*                                                                                 */
/*  Return values    :  ulRC return code from SimLibLogon call                     */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*                                                                                 */
/*      1.  Initialize SimLibLogon parameters.                                     */
/*      2.  Establish a session with the Folder Manager                            */
/*          by calling SimLibLogon.                                                */
/*      3.  Return to main() with Session Handle                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG Logon (PRCSTRUCT pRC, PSZ pszDBName,
             PSZ pszApplicationName, PSZ pszUserID, PSZ pszPassword) {

    /*************************************************************************/
    /* Initialize SimLibLogon parameters.                                    */
    /*************************************************************************/

    RCStruct.ulRC      = SIM_RC_OK;

    /*************************************************************************/
    /* Call SimLibLogon to establish a Folder Manager session.               */
    /* If successful, save the Folder Manager session handle for subsequent  */
    /* Folder Manager API calls.                                             */
    /*************************************************************************/

    SimLibLogon( (PSZ) DBName,            /* Pointer to Database name        */
                 (PSZ) ApplicationName,   /* Pointer to Application Name     */
                 (PSZ) UserID,            /* Pointer to User Id              */
                 (PSZ) Password,          /* Pointer to Password for User Id */
                 (PSZ) NULL,              /* New password */
                 (PSZ) NULL,              /* Logon without proxy             */
                 (PSZ) NULL,              /* Logon without proxy scope       */
                 SIM_SS_NORMAL,           /* Non-configuration session logon */
                 (PASYNCCTLSTRUCT) NULL,  /* Request synchronous processing  */
                 (PRCSTRUCT) &RCStruct ); /* Pointer to RC data structure    */

    if ( RCStruct.ulRC == SIM_RC_OK) {

        hSession = RCStruct.ulParam1;              /* Save Session Handle   */
                                                   /* Save ptr to USERLOGONINFOSTRUCT */
        ulMemoryBlock = RCStruct.ulParam2;         /* for use with SimLibFree */
    }
    return( RCStruct.ulRC );
}

```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : GetDocumentTOC                                              */
/*                                                                                 */
/*  Description      : This function is called from main() to retrieve             */
/*                     the table of contents for a document.  The first            */
/*                     document part will be displayed.                            */
/*                                                                                 */
/*  Function type    :  ULONG.                                                     */
/*                                                                                 */
/*  Input Parameters :  hSession, *RCStruct                                        */
/*                                                                                 */
/*  Output Parameters:  None.                                                      */
/*                                                                                 */
/*  Synopsis         :  ULONG GetDocumentTOC ( hSession, *RCStruct ).              */
/*                                                                                 */
/*  Return values    :  The return code from SimLibGetItemAffiliatedTOC.           */
/*                                                                                 */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*      1.  Call the SimLibGetItemAffiliatedTOC Folder Manager API.                */
/*      2.  Extract and save object data of first object in document.              */
/*      3.  Return to main() with a return code.                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG GetDocumentTOC ( HSESSION hSession, PRCSTRUCT pRC ) {

    PAFFTOCENTRYSTRUCT DocuTOC;           /* TOC data structure            */
    USHORT             count;             /* Loop counter                  */

    /*************************************************************************/
    /* Call SimLibGetItemAffiliatedTOC to get the table of contents for      */
    /* a document.                                                           */
    /* If parts exist for the document, save the pointer to the array of     */
    /* AFFTOCENTRYSTRUCT structures to allow memory to be freed later with   */
    /* SimLibFree.                                                           */
    /*************************************************************************/

    SimLibGetItemAffiliatedTOC(hSession, /* Folder Manager session handle    */
         (PITEMID) DocItemID,           /* Get TOC for this Item ID         */
         SIM_BASE,                      /* Define object type as base       */
         (PASYNCCTLSTRUCT) NULL,        /* Request synchronous processing   */
         (PRCSTRUCT) &RCStruct );       /* Pointer to RC data structure     */

    if (RCStruct.ulRC == SIM_RC_OK) {
        if ((PAFFTOCENTRYSTRUCT)RCStruct.ulParam2 == (PAFFTOCENTRYSTRUCT)NULL) {
            fPartExists = FALSE;
            IpsMessage(strcat(PSZMSGLOGONFAIL, (PSZ)DocItemID), RCStruct.ulRC);
        }
        else {
            ulMemoryBlock = RCStruct.ulParam1;
            fPartExists = TRUE;
            DocuTOC = (PAFFTOCENTRYSTRUCT)RCStruct.ulParam1;

        /*************************************************************************/
        /* Save the Obj structure of the first part associated with the          */
        /* document as the part that will be displayed.                          */
        /*************************************************************************/
            count = 0;                      /* First Object (Document Part) */
            Obj = DocuTOC[count].Obj;       /* save what we need            */
        }
    }

    return( RCStruct.ulRC );
}
```

```
498
499  /*BEGINPROLOGUE*********************************************************************/
500  /*                                                                                  */
501  /*  Function         : QueryDocument                                                */
502  /*                                                                                  */
503  /*  Description      : This function is called from main() to determine the         */
504  /*                     content class of the document to be displayed.               */
505  /*                                                                                  */
506  /*  Function type    :  ULONG.                                                      */
507  /*                                                                                  */
508  /*  Input Parameters :  hSession, *RCStruct                                         */
509  /*                                                                                  */
510  /*  Output Parameters:  None.                                                       */
511  /*                                                                                  */
512  /*  Synopsis         :  ULONG QueryDocument  ( hSession, *RCStruct ).               */
513  /*                                                                                  */
514  /*  Return values    :  Return code from SimLibQueryObject.                         */
515  /*                                                                                  */
516  /*  Process Flow     :                                                              */
517  /*      1.  Call SimLibQueryObject Folder Manager API using the Obj                 */
518  /*          structure returned from SimLibGetItemAffiliatedTOC.                     */
519  /*      2.  Return to main() with a return code.                                    */
520  /*                                                                                  */
521  /*ENDPROLOGUE***********************************************************************/
522
523  ULONG QueryDocument ( HSESSION hSession, PRCSTRUCT pRC ) {
524
525      POBJINFOSTRUCT ObjInfo;             /* Object information structure      */
526
527      /*************************************************************************/
528      /* Call SimLibQueryObject to find the content class of the object.       */
529      /* Save the pointer to the OBJINFOSTRUCT structure so that the memory    */
530      /* can be freed later using SimLibFree.                                  */
531      /*************************************************************************/
532
533      SimLibQueryObject(hSession,              /* Folder Mgr session handle   */
534                        (HOBJ) &Obj,           /* ItemID of object to query   */
535                        (PASYNCCTLSTRUCT)NULL,  /* Request synch. processing   */
536                        (PRCSTRUCT) &RCStruct );/* Ptr to RC data structure    */
537
538      if (RCStruct.ulRC == SIM_RC_OK) {
539          ulMemoryBlock = RCStruct.ulParam1;  /* Save ptr for later SimLibFree */
540          ObjInfo       =  (POBJINFOSTRUCT)RCStruct.ulParam1;
541          ulObjConCls   =   ObjInfo->ulObjConCls;
542      }
543
544      return( RCStruct.ulRC );           /* Return code from SimLibQueryObject */
545  }
546
```

```
547  /*BEGINPROLOGUE*********************************************************/
548  /*                                                                      */
549  /*  Function         : DisplayObject                                    */
550  /*                                                                      */
551  /*  Description      : This function is called from main() to perform the*/
552  /*                     functions necessary to display the first object in*/
553  /*                     a document.                                      */
554  /*                                                                      */
555  /*  Function type    :  ULONG.                                          */
556  /*                                                                      */
557  /*  Input Parameters :  None                                            */
558  /*                                                                      */
559  /*  Output Parameters:  None.                                           */
560  /*                                                                      */
561  /*  Synopsis         :  ULONG DisplayObject ( &RCSTRUCT )               */
562  /*                                                                      */
563  /*  Return values    :  If all functions are successful, SIM_RC_OK is   */
564  /*                      returned.  Otherwise the return code from       */
565  /*                      SimOpsInitEnv, SimWsLoadObj, SimDspCreateWin,   */
566  /*                      or SimWsDeleteObj is returned.                  */
567  /*                                                                      */
568  /*  Process Flow     :                                                  */
569  /*      1.  Initialize the Image Services Environment.                  */
570  /*      2.  Load the document into the working set.                     */
571  /*      3.  Display the first (or only) object in the document.         */
572  /*      4.  Return to main() with a return code.                        */
573  /*                                                                      */
574  /*ENDPROLOGUE***********************************************************/
575
576  ULONG DisplayObject ( PRCSTRUCT pRC ) {
577
578      /**********************************************************************/
579      /*  Make sure RCStruct initialized properly                           */
580      /**********************************************************************/
581
582      RCStruct.ulRC     = SIM_RC_OK;
583      RCStruct.ulStruct = sizeof(RCSTRUCT);    /* must initialize length field */
584
585      /**********************************************************************/
586      /* Call SimOpsInitEnv to initialize the environment.                  */
587      /*   - parameter values defined in IP3DISP.H                          */
588      /**********************************************************************/
589
590      if (SimOpsInitEnv(PSZAPPLDESC,       /* Application description          */
591                  FALSE,                  /* Is this instance unattended?     */
592                  PSZWORKINGDIR,          /* Working directory for application*/
593                  PSZEXITDIR,             /* User exit directory              */
594                  NULL,
595                  PSZOMANAGER,            /* Folder Mgr Object Manager DLL    */
596                  NULL,
597                  (PRCSTRUCT) &RCStruct )   /* Pointer to RC data structure  */
598      |           != SIM_RC_OK ) {
599      |   IpsMessage(PSZMSGOPSINITFAIL, RCStruct.ulRC);  /* Display Error Msg  */
600      |   return( RCStruct.ulRC );
601      |}
602
603      /**********************************************************************/
604      /* Initialize SimWsLoadObj parameters.                                */
605      /**********************************************************************/
606
607      /****************************************/
608      /* SIMCONTROL_BASE structure.           */
609      /****************************************/
610
611      pControl.ulStruct     = sizeof(SIMCONTROL_BASE);
612      pControl.fReserved    = FALSE;
```

```
613     pControl.pszDesc      = NULL;
614
615     /***********************************/
616     /* All attributes are visible.     */
617     /***********************************/
618
619     pControl.ulHideFrom   = SIMUSER_NONE;
620     pControl.ulDocControl = 0;
621
622     /***************************************/
623     /* SIMSRCSERVEROBJ structure.          */
624     /***************************************/
625
626     pDataSrc.hSession     = hSession;
627     pDataSrc.Obj          = Obj;
628     pDataSrc.ulAccess     = SIM_ACCESS_SHARED_READ;
629     pDataSrc.ulPriority   = OM_NORMAL_PRIORITY;
630
631     /*************************************************************************/
632     /* Call SimWsLoadObj to load a working set object.                       */
633     /*************************************************************************/
634
635     if (SimWsLoadObj(SIMTYPE_BASE,    /* Load object as base document       */
636           (USHORT)ulObjConCls,        /* Document content class             */
637           SIMREF_NONE,                /* Load as 1st object in working set  */
638           (SIMWSOBJ)NULL,             /* Handle to working set object       */
639           (PVOID)&pControl,           /* Load control information           */
640           SIMSRC_SERVER_OBJ,          /* Document read from object server   */
641           (PVOID)&pDataSrc,           /* Server object access description   */
642           (PSIMWSOBJ)&hWsObj,         /* Handle of working set object loaded */
643           (PSIMLOAD)&pLoadID,         /* Pointer to load handle             */
644           (PRCSTRUCT) &RCStruct )     /* Pointer to RC data structure       */
645 ┌             != SIM_RC_OK ) {
646 │     IpsMessage(PSZMSGWSLOADOBJFAIL, RCStruct.ulRC); /* Display Error Msg */
647 │     return( RCStruct.ulRC );
648 └}
649
650     /*************************************************************************/
651     /* Initialize SimDspCreateWin parameters.                                */
652     /*************************************************************************/
653
654     /*************************************************************************/
655     /* Create the window with all standard frame features.                   */
656     /*************************************************************************/
657
658     ulCreateFlags = SIMDSP_FRAME_BORDER |
659                     SIMDSP_FRAME_MINMAX |
660                     SIMDSP_FRAME_RESIZE |
661                     SIMDSP_FRAME_SYSTEM |
662                     SIMDSP_FRAME_TITLE;
663
664     /*************************************************************************/
665     /* Specify the window origin.  The origin is taken from the top/left.    */
666     /*************************************************************************/
667
668     PosSize.ulStruct               =  sizeof(SIMPOSSIZE);
669     PosSize.uomType                =  SIMUOM_PIXEL;
670     PosSize.ratOriginX.lNumerator  =  0L;
671     PosSize.ratOriginX.lDenominator =  1;
672     PosSize.ratOriginY.lNumerator  =  0L;
673     PosSize.ratOriginY.lDenominator =  1;
674     PosSize.ratWidth.lNumerator    =  400;
675     PosSize.ratWidth.lDenominator  =  1;
676     PosSize.ratHeight.lNumerator   =  600;
677     PosSize.ratHeight.lDenominator  =  1;
678
```

```
679     /*************************************************************************/
680     /* Setup window title bar.                                               */
681     /*************************************************************************/
682
683     strcpy (szTitle, PSZTITLEBARTEXT);
684     strcat (szTitle, DocItemID);
685
686     /*************************************************************************/
687     /* Call SimDspCreateWin to create a new display window in order to       */
688     /* view the working set data.                                            */
689     /*************************************************************************/
690
691     if (SimDspCreateWin(
692           HWND_DESKTOP,              /* Parent window is the desktop        */
693           HWND_DESKTOP,              /* Owner window is the desktop         */
694           0,                       /* pointer to Icon */
695           szTitle,                   /* Display window title                */
696           (SIMWSOBJ)hWsObj,          /* Working set object handle           */
697           (PSIMPOSSIZE)&PosSize,     /* Window's initial position and size  */
698           (ULONG)ulCreateFlags,      /* Window frame controls               */
699           FALSE,                     /* Create window invisible             */
700           (PSIMWIN)&hWin,            /* Pointer to display window handle    */
701           (PRCSTRUCT) &RCStruct )    /* Pointer to RC data structure        */
702     ┌             != SIM_RC_OK ) {
703     │     IpsMessage(PSZMSGDISPOBJECTFAIL, RCStruct.ulRC);  /* Display Err Msg */
704     └}
705
706     /*************************************************************************/
707     /* Define the Dynamic Information Line for the Display Window            */
708     /*************************************************************************/
709
710     if (SimDspSetWin( hWin,
711                       SIMDSP_DIL,
712                       (PVOID) &pszDilFmt,
713                       &RCStruct )
714     ┌             != SIM_RC_OK ) {
715     │     IpsMessage(PSZMSGDISPSETDILFAIL, RCStruct.ulRC);  /* Display Err Msg */
716     └}
717     if (SimDspSetWin( hWin,
718                       SIMDSP_SHOW,
719                       (PVOID) &fSimDsp,
720                       &RCStruct )
721     ┌             != SIM_RC_OK ) {
722     │     IpsMessage(PSZMSGDISPSHOWFAIL, RCStruct.ulRC);    /* Display Err Msg */
723     └}
724
725     return( RCStruct.ulRC );
726 └}
727
```

```
728  /*BEGINPROLOGUE*********************************************************************/
729  /*                                                                                   */
730  /*  Function         : Cleanup                                                       */
731  /*                                                                                   */
732  /*  Description      : Clean up resources before program terminates.                 */
733  /*                                                                                   */
734  /*  Function type    :  VOID.                                                        */
735  /*                                                                                   */
736  /*  Input Parameters :  ( &RCSTRUCT )                                                */
737  /*                                                                                   */
738  /*  Synopsis         :  VOID Cleanup    (&RCSTRUCT)                                  */
739  /*                                                                                   */
740  /*ENDPROLOGUE***********************************************************************/
741
742  VOID Cleanup    ( PRCSTRUCT pRC ) {
743      if ( hWsObj ) {                       /* Working Set Object handle       */
744          SimWsDeleteObj((SIMWSOBJ)hWsObj,  /* Handle to working set object    */
745               SIMPURGE_ALL,                /* Delete entire working set       */
746               (PRCSTRUCT) &RCStruct );     /* Pointer to RC data structure    */
747      }
748      if ( hSession ) {                          /* Library Manager Session   */
749          SimLibLogoff( hSession,                  /* Folder Mgr session handle */
750                        (PASYNCCTLSTRUCT)NULL,     /* Request synch. processing */
751                        (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure  */
752      }
753      if ( hwnd ) {                              /* PM Window                 */
754          WinDestroyWindow( hwnd );                 /* Destroy PM Window         */
755      }
756      if ( hmq ) {                               /* PM Message Queue          */
757          WinDestroyMsgQueue( hmq );                /* Destroy the message queue */
758      }
759      if ( hab ) {                               /* PM                        */
760          WinTerminate( hab );                    /* Terminate PM environment  */
761      }
762      return;
763  }
764
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/*  Function         : IpsMessage                                                   */
/*                                                                                  */
/*  Description      : Service function to handle application messages.             */
/*                                                                                  */
/*  Function type    :  VOID.                                                       */
/*                                                                                  */
/*  Input Parameters :  (PSZ) Message Text, (ULONG) Error Code                      */
/*                                                                                  */
/*  Output Parameters:  None.                                                       */
/*                                                                                  */
/*  Synopsis         :  VOID IpsMessage (PSZ, ULONG)                                */
/*                                                                                  */
/*  Return values    :  None.                                                       */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

VOID IpsMessage ( PSZ pszMsgText, ULONG ulRC ) {

    char MsgText[80];     /* to format message text */
    sprintf(MsgText, "%s %d\n", pszMsgText, ulRC);

    /* for PM environment */

    WinMessageBox (HWND_DESKTOP,
                   HWND_DESKTOP,
                   MsgText,
                   "Information",
                   0,
                   MB_OK          |
                   MB_INFORMATION |
                   MB_APPLMODAL);

    /* for command line environment */

 /* printf(MsgText); */

    return;
}

/*************************************************************************************/
/* End of IP3DISP.C                                                                  */
/*************************************************************************************/
```

D.4 IP3DISP.DEF (11/04/93 09:03:28)

```
;*BEGINPROLOGUE*********************************************************************/
;*                                                                                  */
;* MODULE: IP3DISP.DEF                                                             */
;*                                                                                  */
;* DISCLAIMER OF WARRANTIES:                                                        */
;* -------------------------                                                       */
;* The following [enclosed] code is sample code created by IBM                     */
;* Corporation.  This sample code is not part of any standard IBM product          */
;* and is provided to you solely for the purpose of assisting you in the           */
;* development of your applications.  The code is provided "AS IS",                */
;* without warranty of any kind.  IBM shall not be liable for any damages          */
;* arising out of your use of the sample code, even if they have been              */
;* advised of the possibility of such damages.                                     */
;*                                                                                  */
;* Status:     Version 1 Release 1 Modification 0                                  */
```

```
;*                                                                            */
;* Description: IP3DISP.DEF is used by program IP3DISP.C.                    */
;*                                                                            */
;*ENDPROLOGUE*****************************************************************/

NAME IP3DISP WINDOWAPI

DESCRIPTION 'Spectacular Bid Sample Application'

STUB    'OS2STUB.EXE'

CODE    MOVEABLE
DATA    MOVEABLE MULTIPLE

HEAPSIZE   64000        ; Must be non-zero to use Local memory manager
STACKSIZE  64000        ; Must be non-zero for SS == DS
                        ; suggest 4k as minimum stacksize
;*
;* End of IP3DISP.DEF
```

D.5 IP3DISP.MAK (04/25/94 10:55:04)

```
#*******************************************************************************
#
# Make file for IP3DISP.C                        ImagePlus VisualInfo
#
#*******************************************************************************
#
#  COPYRIGHT:
#  ----------
#  Copyright (C) International Business Machines Corp., 1993, 1994.
#
#
#  DISCLAIMER OF WARRANTIES:
#  -------------------------
#  The following [enclosed] code is sample code created by IBM
#  Corporation.  This sample code is not part of any standard IBM product
#  and is provided to you solely for the purpose of assisting you in the
#  development of your applications.  The code is provided "AS IS",
#  without warranty of any kind.  IBM shall not be liable for any damages
#  arising out of your use of the sample code, even if they have been
#  advised of the possibility of such damages.
#
#*******************************************************************************
#
# External Environment Variables Used
#
#   SB_ROOT       This environment variable defines where the
#                 VisualInfo directory is located.  It maybe set
#                 either in the config.sys file or from the
#                 command line. For example,
#
#                     SET SB_ROOT=D:\FRNV1R0
#
#   DEBUG         This environment variable defines whether a
#                 Debug or Non-Debug build is to be performed.
#                 It maybe set either in the config.sys file or
#                 from the command line(when switching back and
#                 forth is desired).  The variable may be set
#                 according to
#
#                     SET DEBUG=1  --> perform Debug build
#
```

```
#                   or,
#
#                   SET DEBUG=0  --> perform Non-Debug build
#
#*****************************************************************************
#
# Compiler Options Used
#
#  /C+          Compile only
#  /Fd-         Store internal work files in shared memory
#  /Gd+         Dynamically link the run-time library
#  /Ge+         Build an EXE file
#  /Ge-         Build a  DLL file
#  /Gh+         Generate code for profiling
#  /Gh-         Disable profiling
#  /Gm+         Link with multi-threaded version of library
#  /I           Include file location
#  /Kbcfogapexir  Give all diagnostics except preprocessor trace
#  /Lf-         Set all listing options off
#  /Mp          Use - optlink - linkage for functions
#  /Q+          Do not display logo
#  /Re          Generate executable code that can be used in an OS/2 runtime
#               environment
#  /Rn          Generate executable code that can be used as a subsystem with
#               no runtime environment
#  /Se          Allow all C Set ++ language extensions except migration
#  /Sm          Control Compiler interpretation of unsupported keywords
#  /Sn+         Allow use of DBCS
#  /Sp1         Align on single byte boundires (i.e pack)
#  /Sp4         Align on full word boundires (i.e don't pack)
#  /Ss          Allow double slash (//) for comments
#  /Ti+         Generate debugging information
#
#  /ALIGN       Set the alignment factor.  Must be power of 2.
#  /CO          Include symbolic debugger info
#  /DE          Prepare for debugging.
#  /EXEPACK     Compress byte pattern in *.exe
#  /MAP         Create *.map file.  Lisat public symbols.
#  /NOI         NOIGNORECASE           identifiers are case sensitive
#  /NOL         NOLOGO                 disable signon banner
#  /NOD         NODEFAULTLIBRARYSEARCH don't search LIB libraries
#  /PACKCODE    Pack neighboring code segments.
#  /PACKDATA    Pack neighboring data segments.
#
#*****************************************************************************

#|"
#] Generic/Common Macros                                                    ]
#Î¬
CCFLAGS  = /C+ /Gd+ /Ge+ /Gm+ /Kb /Mp /Q+ /Re /Se /Sn+ /Sp1 /Ss

INCLUDES = $(SB_ROOT)\INCLUDE;$(SB_ROOT)\INC

LLFLAGS  = /NOI /NOL

#|"
#] Set up defaults                                                          ]
#Î¬

!ifndef DEBUG
DEBUG=0
!endif

#|"
#] Debug/Non-Debug Macros                                                   ]
#Î¬
```

```
!if $(DEBUG)
CFLAGS   =  /Gh- /Lf- /Ti+ $(CCFLAGS) /I $(INCLUDES)
LFLAGS   = /CO /DE $(LLFLAGS)
BINDIR   = .
LIBDIR   = $(SB_ROOT)\LIB
SRCDIR   = .
OBJDIR   = .
SRCFIL   = IP3DISP
!else
CFLAGS   = $(CCFLAGS) /I $(INCLUDES)
LFLAGS   = $(LLFLAGS)
BINDIR   = .
LIBDIR   = $(SB_ROOT)\LIB
SRCDIR   = .
OBJDIR   = .
SRCFIL   = IP3DISP
!endif

COMPILE_REG = ICC $(CFLAGS) -Fo$(OBJDIR)\$@ $(SRCDIR)\$(@B).C

LINKLIBS = $(LIBDIR)\FRNOFI.LIB    \
           $(LIBDIR)\FRNOFO.LIB    \
           $(LIBDIR)\FIWSENV.LIB   \
           $(LIBDIR)\FIWSP.LIB     \
           $(LIBDIR)\FIWSWS.LIB    \
           $(LIBDIR)\FIWSD.LIB     \
           OS2386.LIB

OFILES   = $(OBJDIR)\$(SRCFIL).OBJ

#├"
#] Dependencies                                                    ]
#Î┐

FOLDERMGR_DEP = {$(INCLUDES);}FRNP.H         {$(INCLUDES);}FRNPCAPI.H  \
                {$(INCLUDES);}FRNPERR.H      {$(INCLUDES);}FRNPFI.H    \
                {$(INCLUDES);}FRNPTYPE.H     {$(INCLUDES);}FRNPFI2.H   \
                {$(INCLUDES);}FRNPLIBC.H     {$(INCLUDES);}FRNPLCLI.H  \
                {$(INCLUDES);}FRNPFO.H

IS_DEP        = {$(INCLUDES);}FIWS.H         {$(INCLUDES);}FIWSDSP.H   \
                {$(INCLUDES);}FIWSWS.H       {$(INCLUDES);}FIWSENV.H

FRNOEXDI_DEP  = {$(INCLUDES);}$(SRCFIL).H     $(FOLDERMGR_DEP)          \
                  $(IS_DEP)

#├"
#] Main Target Rule                                                ]
#Î┐
ALL         : ERASE $(BINDIR)\$(SRCFIL).EXE

ERASE       :
    if exist ERR.LST erase ERR.LST

#├"
#] Rules                                                           ]
#Î┐
$(BINDIR)\$(SRCFIL).EXE   :  $(OFILES) $(SRCDIR)\$(SRCFIL).DEF $(SRCFIL).MAK
   LINK386 $(LFLAGS) $(OFILES),    \
           $(BINDIR)\$(SRCFIL).EXE, \
           $(BINDIR)\$(SRCFIL).MAP, \
           $(LINKLIBS),            \
           $(SRCDIR)\$(SRCFIL).DEF

$(OBJDIR)\$(SRCFIL).OBJ : $(SRCDIR)\$(SRCFIL).C $(FRNOEXDI_DEP)
   $(COMPILE_REG)
```

Appendix E. IP3PRINT Source Code

E.1 Function index

GetDocumentTOC (IP3PRINT.C, page 147)	
calls	**IpsMessage** (IP3PRINT.C, page 152) **SimLibGetItemAffiliatedTOC** **strcat**
called by	**main** (IP3PRINT.C, page 143)

IpsMessage (IP3PRINT.C, page 152)	
calls	**sprintf** **WinMessageBox**
called by	**GetDocumentTOC** (IP3PRINT.C, page 147) **main** (IP3PRINT.C, page 143) **PrintDocument** (IP3PRINT.C, page 149)

Logon (IP3PRINT.C, page 146)	
calls	**SimLibLogon**
called by	**main** (IP3PRINT.C, page 143)

main (IP3PRINT.C, page 143)	
calls	**GetDocumentTOC** (IP3PRINT.C, page 147) **IpsMessage** (IP3PRINT.C, page 152) **Logon** (IP3PRINT.C, page 146) **PrintDocument** (IP3PRINT.C, page 149) **QueryDocument** (IP3PRINT.C, page 148) **SimLibFree** **SimLibLogoff** **strcpy** **strncpy** **WinCreateMsgQueue** **WinInitialize**

PrintDocument (IP3PRINT.C, page 149)	
calls	**IpsMessage** (IP3PRINT.C, page 152) **SimOpsInitEnv** **SimPrtPrintObject** **SimWsDeleteObj** **SimWsLoadObj**
called by	**main** (IP3PRINT.C, page 143)

QueryDocument (IP3PRINT.C, page 148)	
calls	**SimLibQueryObject**
called by	**main** (IP3PRINT.C, page 143)

SimLibFree	
called by	**main** (IP3PRINT.C, page 143)

SimLibGetItemAffiliatedTOC	
called by	**GetDocumentTOC** (IP3PRINT.C, page 147)

SimLibLogoff	
called by	**main** (IP3PRINT.C, page 143)

SimLibLogon	
called by	**Logon** (IP3PRINT.C, page 146)

SimLibQueryObject	
called by	**QueryDocument** (IP3PRINT.C, page 148)

SimOpsInitEnv	
called by	**PrintDocument** (IP3PRINT.C, page 149)

SimPrtPrintObject	
called by	**PrintDocument** (IP3PRINT.C, page 149)

SimWsDeleteObj	
called by	**PrintDocument** (IP3PRINT.C, page 149)

SimWsLoadObj	
called by	**PrintDocument** (IP3PRINT.C, page 149)

sprintf	
called by	**IpsMessage** (IP3PRINT.C, page 152)

strcat	
called by	**GetDocumentTOC** (IP3PRINT.C, page 147)

strcpy	
called by	**main** (IP3PRINT.C, page 143)

strncpy	
called by	**main** (IP3PRINT.C, page 143)

WinCreateMsgQueue	
called by	**main** (IP3PRINT.C, page 143)

WinInitialize	
called by	**main** (IP3PRINT.C, page 143)

WinMessageBox	
called by	**IpsMessage** (IP3PRINT.C, page 152)

E.2 IP3PRINT.H (05/20/94 14:58:20)

```
1  /*****************************************************************************/
2  /*                                                                           */
3  /* MODULE: IP3PRINT.H                                                        */
4  /*                                                                           */
5  /* Description  : Header file for module IP3PRINT.C.                         */
6  /*                                                                           */
7  /*****************************************************************************/
8
9  /*****************************************************************************/
10 /* IP3PRINT.C Function Prototypes                                            */
11 /*****************************************************************************/
12
13 ULONG    Logon              (PRCSTRUCT pRC,
14                              PSZ pszDBName,
15                              PSZ pszApplicationName,
16                              PSZ pszUserID,
17                              PSZ pszPassword) ;
18 ULONG    GetDocumentTOC      (HSESSION hSession,
19                              PRCSTRUCT pRC) ;
20 ULONG    QueryDocument       (HSESSION hSession,
21                              PRCSTRUCT pRC) ;
22 ULONG    PrintDocument       (PRCSTRUCT pRC) ;
23 VOID     IpsMessage          (PSZ   pszMsgText,
24                              ULONG ulRC) ;
25
26 /*****************************************************************************/
27 /* Presentation Manager Definitions                                          */
28 /*****************************************************************************/
29
30 #define INCL_WINWINDOWMGR
31
32 /*****************************************************************************/
33 /* SimLibLogon Parameter Definitions                                         */
34 /*****************************************************************************/
35
36 #define PSZDBNAME          "LIBSRVR2"
37 #define PSZAPPLICATIONNAME "SAMPLE"
38 #define PSZUSERID          "FRNADMIN"
39 #define PSZPASSWORD        "PASSWORD"
40
41 /*****************************************/
42 /* SimLibLogon Parameter Maximum Lengths */
43 /*****************************************/
```

```
44
45  #define MAXDBNAME           18          /* Maximum size of pszDBName         */
46  #define MAXAPPLICATIONNAME  8           /* Maximum size of pszApplicationName*/
47  #define MAXUSERID           26          /* Maximum size of pszUserID         */
48  #define MAXPASSWORD         8           /* Maximum size of pszPassword       */
49
50  /*****************************************************************************/
51  /* SimOpsInitEnv Parameter Definitions                                       */
52  /*****************************************************************************/
53
54  #define PSZAPPLDESC         "Sample scenario to print a document."
55  #define PSZWORKINGDIR       "D:\\FRNV1RO\\WORK"
56  #define PSZEXITDIR          "D:\\FRNV1RO\\DLL"
57  #define PSZOIMANAGER        "FRNOFI"
58  #define PSZOMANAGER         "FRNOFO"
59
60  /*******************************************/
61  /* SimOpsInitEnv Parameter Maximum Lengths */
62  /*******************************************/
63
64  #define MAXDIRNAME           255        /* Maximum size of pszWorkingDir,    */
65                                          /* pszExitDir, pszOIManager,         */
66                                          /* and pszOManager                   */
67
68  /*************************************************/
69  /* SimOpsInitEnv and SimWsLoadObj Maximum Lengths */
70  /*************************************************/
71
72  #define MAXDESC              40         /* Maximum size of pszApplDesc       */
73                                          /* and pszDocDes                     */
74
75  /*****************************************************/
76  /* Printer Option Profile Name for SimPrtPrintObject */
77  /*****************************************************/
78
79  #define PSZOS2PRTNAME     "SAMPLEAPPRT"
80
81
82  /*******************************************/
83  /* Message Texts                           */
84  /*******************************************/
85
86  #define PSZMSGPARMERR       "DocID only is required.  Number of parms passed ="
87  #define PSZMSGPRTFAIL       "Print Document Error.  RC ="
88  #define PSZMSGQRYDOCFAIL    "Query Document error.  RC ="
89  #define PSZMSGQRYTOCFAIL    "Query Document TOC error.  RC ="
90  #define PSZMSGLOGONFAIL     "Library Logon error.  RC ="
91  #define PSZMSGNODOCPART     "There are no parts for document "
92  #define PSZMSGOPSINITFAIL   "SimOpsInitEnv error.  RC ="
93  #define PSZMSGWSLOADOBJFAIL "SimWsLoadObj error.  RC ="
94  #define PSZMSGPRINTOBJECTFAIL "SimPrtPrintObject error.  RC ="
95  #define PSZMSGPRINTOK       "Print request completed  RC ="
96
97  /*****************************************************************************/
98  /* End of IP3PRINT.H                                                         */
99  /*****************************************************************************/
```

E.3 IP3PRINT.C (06/28/94 14:06:32)

GetDocumentTOC (IP3PRINT.C, page 147)	
calls	**IpsMessage** (IP3PRINT.C, page 152) **SimLibGetItemAffiliatedTOC** **strcat**
called by	**main** (IP3PRINT.C, page 143)

IpsMessage (IP3PRINT.C, page 152)	
calls	**sprintf** **WinMessageBox**
called by	**GetDocumentTOC** (IP3PRINT.C, page 147) **main** (IP3PRINT.C, page 143) **PrintDocument** (IP3PRINT.C, page 149)

Logon (IP3PRINT.C, page 146)	
calls	**SimLibLogon**
called by	**main** (IP3PRINT.C, page 143)

main (IP3PRINT.C, page 143)	
calls	**GetDocumentTOC** (IP3PRINT.C, page 147) **IpsMessage** (IP3PRINT.C, page 152) **Logon** (IP3PRINT.C, page 146) **PrintDocument** (IP3PRINT.C, page 149) **QueryDocument** (IP3PRINT.C, page 148) **SimLibFree** **SimLibLogoff** **strcpy** **strncpy** **WinCreateMsgQueue** **WinInitialize**

PrintDocument (IP3PRINT.C, page 149)	
calls	**IpsMessage** (IP3PRINT.C, page 152) **SimOpsInitEnv** **SimPrtPrintObject** **SimWsDeleteObj** **SimWsLoadObj**
called by	**main** (IP3PRINT.C, page 143)

QueryDocument (IP3PRINT.C, page 148)	
calls	**SimLibQueryObject**
called by	**main** (IP3PRINT.C, page 143)

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/* MODULE: IP3PRINT.C                                                               */
/*                                                                                  */
/* DISCLAIMER OF WARRANTIES:                                                        */
/* -------------------------                                                        */
/* The following [enclosed] code is sample code created by IBM                      */
/* Corporation.  This sample code is not part of any standard IBM product           */
/* and is provided to you solely for the purpose of assisting you in the            */
/* development of your applications.  The code is provided "AS IS",                 */
/* without warranty of any kind.  IBM shall not be liable for any damages           */
/* arising out of your use of the sample code, even if they have been               */
/* advised of the possibility of such damages.                                      */
/*                                                                                  */
/* Description : IP3PRINT.c - Sample application code to print a document.          */
/*               Folder Manager APIs used are : SimLibLogon, SimLibFree,            */
/*               SimLibQueryObject and SimLibLogoff.                                */
/*               Image Services APIs used are: SimOpsInitEnv,                       */
/*               SimWsLoadObj, SimPrtPrintObject and SimWsDeleteObj.                */
/*                                                                                  */
/* System      : OS/2 compatible PC                                                 */
/* OS          : OS/2 2.1                                                           */
/* Compiler    : IBM C/C++                                                          */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

/************************************************************************************/
/*  System Header Files                                                             */
/************************************************************************************/
#include <os2.h>                        /* Main OS/2 header files                   */
#include <stdio.h>                      /* Standard I/O header files                */
#include <stdlib.h>                     /* Standard Library header files            */
#include <string.h>                     /* String manipulation                      */

/************************************************************************************/
/*  ImagePlus - Folder Manager header files                                         */
/************************************************************************************/
#define   FRN_INCL_FI                   /* Include frnpfi.h & frnplcli.h            */
#define   FRN_INCL_FO                   /* Include frnpfo.h                         */
#include "frnp.h"                       /* Global types                             */
#include "frnpcapi.h"                   /* Structures and constants                 */

/************************************************************************************/
/*  ImagePlus - Image Services Header Files                                         */
/************************************************************************************/
#define   FIWS_SERVER                   /* Required to read from a server           */
#include "fiws.h"                       /* Base Image Services                      */
#include "fiwsenv.h"                    /* Environment prototypes                   */
#include "fiwsws.h"                     /* Working Set prototypes                   */
#include "fiwsprt.h"                    /* Print prototypes                         */
#include "fiwsdev.h"                    /* Common Device Include - JR               */
/************************************************************************************/
/*  IP3PRINT Header File                                                            */
/************************************************************************************/
#include "IP3PRINT.h"                   /* Application header file                  */

/************************************************************************************/
/*  Global Variables                                                                */
/************************************************************************************/
/*  OS/2 and PM APIs                                                                */
/************************************************************************************/
HAB       hab;                          /* Anchor block handle                      */
HMQ       hmq;                          /* Message Queue handle                     */

/************************************************************************************/
/*  ImagePlus APIs - Library Session data used with Folder Manager APIs             */
```

```
67  /*****************************************************************************/
68  HSESSION hSession;                      /* Folder Manager session handle     */
69  RCSTRUCT RCStruct;                      /* RC data structure for API calls   */
70  ULONG    ulMemoryBlock;                 /* Generic pointer for SimLibFree    */
71  CHAR     DBName[MAXDBNAME + 1] = "";              /* Library Server Name    */
72  CHAR     ApplicationName[MAXAPPLICATIONNAME +1] = ""; /* Application Name    */
73  CHAR     UserID[MAXUSERID +1] = "";               /* User ID               */
74  CHAR     Password[MAXPASSWORD + 1] = "";          /* Password              */
75  ITEMID   DocItemID;                     /* Document to print                 */
76  BOOL     fPartExists;                   /* Does a part exist for the document*/
77  OBJ      Obj;                           /* Document part to print            */
78  ULONG    ulObjConCls;                   /* Document content class            */
79
80  /*****************************************************************************/
81  /*  ImagePlus APIs - Image Services Data - used with Image Services APIs     */
82  /*****************************************************************************/
83  SIMWSOBJ hWsObj;                        /* Working set object handle         */
84
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : main                                                        */
/*                                                                                 */
/*  Description      : main() function of sample program to print a document       */
/*                                                                                 */
/*  Function type    :  INT.                                                       */
/*                                                                                 */
/*  Input Parameters :  DocItemID.                                                 */
/*                      e.g. c:>IP3PRINT W1263629M0511777                          */
/*                                                                                 */
/*  Output Parameters:  None.                                                      */
/*                                                                                 */
/*  Synopsis         :  INT main ( int argc, char *argv[] )                        */
/*                                                                                 */
/*  Return values    :  If all functions are successful, the return code           */
/*                      from Logoff() is returned.  Otherwise, the return          */
/*                      code from the failing function is returned.                */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*      1.  Initialize Presentation Manager.                                       */
/*      2.  Establish a session with the Folder Manager                            */
/*      3.  If base objects exist for the document:                                */
/*          objects exist for the document :                                       */
/*          - Save the Obj structure of the first document part as                 */
/*            the document that will be printed.  Free the memory                  */
/*            containing the array of PAFFTOCENTRYSTRUCT structures                */
/*          - Query the object to obtain the document content class               */
/*          - Free the memory containing the OBJINFOSTRUCT structure.             */
/*      4.  Print the document using Image Services                                */
/*      5.  Logoff from the Folder Manager session.                                */
/*      6.  Exit main() with a return code.                                        */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

INT main (int argc, char *argv[]) {

    ULONG ulretcode = 0;

    /*************************************************************************/
    /* Initialize Session Variables required for logon.                      */
    /*************************************************************************/
    strcpy (DBName,           PSZDBNAME);
    strcpy (ApplicationName,  PSZAPPLICATIONNAME);
    strcpy (UserID,           PSZUSERID);
    strcpy (Password,         PSZPASSWORD);

    /*************************************************************************/
    /* Initialize PM, and create Message Queue so we can use WinMessageBox.  */
    /* PM is required for Image Services "PRINT" function to operate.        */
    /*************************************************************************/

    hab = WinInitialize((USHORT) NULL);
    hmq = WinCreateMsgQueue( hab, 0 );
    if (hab == NULLHANDLE) {
        return(99);                                 /* Quit Immediately      */
    }

    /*************************************************************************/
    /* Check that Document ID has been passed as parameter.                  */
    /*************************************************************************/

    if (argc == 2) {                        /* Program Name is always 1st arg */
        strncpy ( (PSZ)DocItemID, (PSZ)argv[1], DOC_ID_SIZE);
```

```
149     }
150     else {
151         IpsMessage(PSZMSGPARMERR, (ULONG) (argc - 1));
152         return(0);
153     }
154
155     /*************************************************************************/
156     /*  Call Logon to establish a Folder Manager Session.                    */
157     /* If successful, call SimLibFree to free the memory for structure       */
158     /* USERLOGONINFOSTRUC returned from SimLibLogon.                         */
159     /*************************************************************************/
160
161     ulretcode = Logon ( &RCStruct,                 /* API Return Code struct*/
162                         DBName,                   /* Library Server name   */
163                         ApplicationName,          /* Application Name      */
164                         UserID,                   /* User ID               */
165                         Password);                /* Password              */
166
167     if (ulretcode == SIM_RC_OK) {
168
169         SimLibFree( hSession,                  /* Folder Mgr session handle  */
170                     (PVOID) ulMemoryBlock,     /* Ptr to memory block to free*/
171                     (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure   */
172     }
173     else {
174         IpsMessage(PSZMSGLOGONFAIL, ulretcode);
175         return(ulretcode);                        /* Quit Immediately      */
176     }
177
178     /*************************************************************************/
179     /*  Call GetDocumentTOC to find the document parts.                      */
180     /*  If successful, and base parts exist for the document, call           */
181     /*  FreeMemory to free the memory for an array of AFFTOCENTRYSTRUCT       */
182     /*  structures returned from SimLibGetItemAffiliatedTOC.                 */
183     /*************************************************************************/
184
185     ulretcode = GetDocumentTOC( hSession, &RCStruct );
186
187     if ( (ulretcode == SIM_RC_OK) && (fPartExists == TRUE) ) {
188         SimLibFree( hSession,                  /* Folder Mgr session handle  */
189                     (PVOID) ulMemoryBlock,     /* Ptr to memory block to free*/
190                     (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure   */
191     }
192     else {
193         IpsMessage(PSZMSGQRYTOCFAIL, ulretcode);
194         SimLibLogoff( hSession,                /* Folder Mgr session handle */
195                       (PASYNCCTLSTRUCT)NULL,   /* Request synch. processing */
196                       (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure  */
197         return(ulretcode);                        /* Quit Immediately      */
198     }
199
200     /*************************************************************************/
201     /*  Call QueryDocument to get the content class of the document part.    */
202     /* If successful, call FreeMemory to free the memory for structure       */
203     /* OBJINFOSTRUCT returned from SimLibQueryObject.                        */
204     /*************************************************************************/
205
206     ulretcode = QueryDocument( hSession, &RCStruct );
207
208     if (ulretcode == SIM_RC_OK) {
209         SimLibFree( hSession,                  /* Folder Mgr session handle  */
210                     (PVOID) ulMemoryBlock,     /* Ptr to memory block to free*/
211                     (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure   */
212     }
```

```
213     else {
214         IpsMessage(PSZMSGQRYDOCFAIL, ulretcode);
215         SimLibLogoff( hSession,                 /* Folder Mgr session handle */
216                       (PASYNCCTLSTRUCT)NULL,    /* Request synch. processing */
217                       (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure  */
218         return(ulretcode);                        /* Quit Immediately      */
219     }
220
221     /*************************************************************************/
222     /* Call PrintDocument to print the document.                             */
223     /*************************************************************************/
224
225     ulretcode = PrintDocument( &RCStruct);
226
227     if (ulretcode != SIM_RC_OK) {
228         IpsMessage(PSZMSGPRTFAIL, ulretcode);
229     }
230
231     /*************************************************************************/
232     /* Logoff Folder Manager and return.                                     */
233     /*************************************************************************/
234
235     ulretcode = SimLibLogoff( hSession,         /* Folder Mgr session handle */
236                   (PASYNCCTLSTRUCT)NULL,        /* Request synch. processing */
237                   (PRCSTRUCT) &RCStruct );      /* Ptr to RC data structure  */
238     return ( ulretcode );
239 }
240
```

```
241  /*BEGINPROLOGUE*********************************************************/
242  /*                                                                      */
243  /*  Function         : Logon                                           */
244  /*                                                                      */
245  /*  Description      : This function is called from main() to logon to */
246  /*                     ImagePlus Bid Folder Manager.                    */
247  /*                                                                      */
248  /*  Function type    :  ULONG                                          */
249  /*                                                                      */
250  /*  Input Parameters :  pszDBName pszApplicationName pszUserID pszPassword */
251  /*                                                                      */
252  /*  Output Parameters:  hSession RCSTRUCT                             */
253  /*                                                                      */
254  /*  Synopsis         :  ULONG Logon ( pRC                              */
255  /*                                    pszDBName,                        */
256  /*                                    pszApplicationName,               */
257  /*                                    pszUserID,                        */
258  /*                                    pszPassword)                      */
259  /*                                                                      */
260  /*  Return values    :  ulRC return code from SimLibLogon call         */
261  /*                                                                      */
262  /*  Process Flow     :                                                 */
263  /*      1.  Initialize SimLibLogon parameters.                          */
264  /*      2.  Establish a session with the Folder Manager                 */
265  /*          by calling SimLibLogon.                                     */
266  /*      3.  Return to main() with Session Handle                        */
267  /*                                                                      */
268  /*ENDPROLOGUE***********************************************************/
269
270  ULONG Logon (PRCSTRUCT pRC, PSZ pszDBName,
271               PSZ pszApplicationName, PSZ pszUserID, PSZ pszPassword) {
272
273      /*************************************************************************/
274      /* Initialize SimLibLogon parameters.                                    */
275      /*************************************************************************/
276
277      RCStruct.ulRC      = SIM_RC_OK;
278
279      /*************************************************************************/
280      /* Call SimLibLogon to establish a Folder Manager session.               */
281      /* If successful, save the Folder Manager session handle for subsequent  */
282      /* Folder Manager API calls.                                             */
283      /*************************************************************************/
284
285      SimLibLogon( (PSZ) DBName,            /* Pointer to Database name        */
286                   (PSZ) ApplicationName,   /* Pointer to Application Name     */
287                   (PSZ) UserID,            /* Pointer to User Id              */
288                   (PSZ) Password,          /* Pointer to Password for User Id */
289                   (PSZ) NULL,              /* Logon without proxy             */
290                   (PSZ) NULL,              /* Logon without proxy scope       */
291                   (PSZ) NULL,
292                   SIM_SS_NORMAL,           /* Non-configuration session logon */
293                   (PASYNCCTLSTRUCT) NULL,  /* Request synchronous processing  */
294                   (PRCSTRUCT) &RCStruct ); /* Pointer to RC data structure    */
295
296      if ( RCStruct.ulRC == SIM_RC_OK) {
297
298          hSession = RCStruct.ulParam1;              /* Save Session Handle   */
299                                                     /* Save ptr to USERLOGONINFOSTRUCT */
300          ulMemoryBlock = RCStruct.ulParam2;         /* for use with SimLibFree */
301      }
302      return( RCStruct.ulRC );
303  }
304
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : GetDocumentTOC                                              */
/*                                                                                 */
/*  Description      : This function is called from main() to retrieve             */
/*                     the table of contents for a document.  The first            */
/*                     document part will be print.                                */
/*                                                                                 */
/*  Function type    :  ULONG.                                                     */
/*                                                                                 */
/*  Input Parameters :  hSession, *RCStruct                                        */
/*                                                                                 */
/*  Output Parameters:  None.                                                      */
/*                                                                                 */
/*  Synopsis         :  ULONG GetDocumentTOC ( hSession, *RCStruct ).              */
/*                                                                                 */
/*  Return values    :  The return code from SimLibGetItemAffiliatedTOC.           */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*      1.  Call the SimLibGetItemAffiliatedTOC Folder Manager API.                */
/*      2.  Extract and save object data of first object in document.              */
/*      3.  Return to main() with a return code.                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG GetDocumentTOC ( HSESSION hSession, PRCSTRUCT pRC ) {

    PAFFTOCENTRYSTRUCT DocuTOC;            /* TOC data structure             */
    USHORT             count;              /* Loop counter                   */

    /*********************************************************************/
    /* Call SimLibGetItemAffiliatedTOC to get the table of contents for  */
    /* a document.                                                       */
    /* If parts exist for the document, save the pointer to the array of */
    /* AFFTOCENTRYSTRUCT structures to allow memory to be freed later with */
    /* SimLibFree.                                                       */
    /*********************************************************************/

    SimLibGetItemAffiliatedTOC(hSession, /* Folder Manager session handle    */
         (PITEMID) DocItemID,           /* Get TOC for this Item ID         */
         SIM_BASE,                      /* Define object type as base       */
         (PASYNCCTLSTRUCT) NULL,        /* Request synchronous processing   */
         (PRCSTRUCT) &RCStruct );       /* Pointer to RC data structure     */

    if (RCStruct.ulRC == SIM_RC_OK) {
        if ((PAFFTOCENTRYSTRUCT)RCStruct.ulParam2 == (PAFFTOCENTRYSTRUCT)NULL) {
            fPartExists = FALSE;
            IpsMessage(strcat(PSZMSGLOGONFAIL, (PSZ)DocItemID), RCStruct.ulRC);
        }
        else {
            ulMemoryBlock = RCStruct.ulParam1;
            fPartExists = TRUE;
            DocuTOC = (PAFFTOCENTRYSTRUCT)RCStruct.ulParam1;

    /*********************************************************************/
    /* Save the Obj structure of the first part associated with the      */
    /* document as the part that will be printed.                        */
    /*********************************************************************/
            count = 0;                      /* First Object (Document Part) */
            Obj = DocuTOC[count].Obj;       /* save what we need            */
        }
    }

    return( RCStruct.ulRC );
}

```

```
371  /*BEGINPROLOGUE*********************************************************************/
372  /*                                                                                   */
373  /*  Function         : QueryDocument                                                 */
374  /*                                                                                   */
375  /*  Description      : This function is called from main() to determine the         */
376  /*                     content class of the document to be printed.                  */
377  /*                                                                                   */
378  /*  Function type    :  ULONG.                                                       */
379  /*                                                                                   */
380  /*  Input Parameters :  hSession, *RCStruct                                          */
381  /*                                                                                   */
382  /*  Output Parameters:  None.                                                        */
383  /*                                                                                   */
384  /*  Synopsis         :  ULONG QueryDocument  ( hSession, *RCStruct ).                */
385  /*                                                                                   */
386  /*  Return values    :  Return code from SimLibQueryObject.                          */
387  /*                                                                                   */
388  /*  Process Flow     :                                                               */
389  /*      1.  Call SimLibQueryObject Folder Manager API using the Obj                  */
390  /*          structure returned from SimLibGetItemAffiliatedTOC.                      */
391  /*      2.  Return to main() with a return code.                                     */
392  /*                                                                                   */
393  /*ENDPROLOGUE***********************************************************************/
394
395  ULONG QueryDocument ( HSESSION hSession, PRCSTRUCT pRC ) {
396
397      POBJINFOSTRUCT ObjInfo;             /* Object information structure      */
398
399      /*************************************************************************/
400      /* Call SimLibQueryObject to find the content class of the object.       */
401      /* Save the pointer to the OBJINFOSTRUCT structure so that the memory    */
402      /* can be freed later using SimLibFree.                                  */
403      /*************************************************************************/
404
405      SimLibQueryObject(hSession,              /* Folder Mgr session handle   */
406                        (HOBJ) &Obj,           /* ItemID of object to query   */
407                        (PASYNCCTLSTRUCT)NULL, /* Request synch. processing   */
408                        (PRCSTRUCT) &RCStruct );/* Ptr to RC data structure   */
409
410      if (RCStruct.ulRC == SIM_RC_OK) {
411          ulMemoryBlock = RCStruct.ulParam1;  /* Save ptr for later SimLibFree */
412          ObjInfo       =  (POBJINFOSTRUCT)RCStruct.ulParam1;
413          ulObjConCls   =   ObjInfo->ulObjConCls;
414      }
415
416      return( RCStruct.ulRC );           /* Return code from SimLibQueryObject */
417  }
418
```

```
419  /*BEGINPROLOGUE*********************************************************************/
420  /*                                                                                   */
421  /*  Function         : PrintDocument                                                 */
422  /*                                                                                   */
423  /*  Description      : This function is called from main() to perform the           */
424  /*                     functions necessary to print the first object in              */
425  /*                     a document.                                                   */
426  /*                                                                                   */
427  /*  Function type    :  ULONG.                                                       */
428  /*                                                                                   */
429  /*  Input Parameters :  *RCStruct                                                    */
430  /*                                                                                   */
431  /*  Output Parameters:  None.                                                        */
432  /*                                                                                   */
433  /*  Synopsis         :  ULONG PrintDocument ( PRCSTRUCT pRC )                        */
434  /*                                                                                   */
435  /*  Return values    :  If all functions are successful, SIM_RC_OK is                */
436  /*                      returned.  Otherwise the return code from                    */
437  /*                      SimOpsInitEnv, SimWsLoadObj, SimPrtPrintObject               */
438  /*                      or SimWsDeleteObj is returned.                               */
439  /*                                                                                   */
440  /*  Process Flow     :                                                               */
441  /*      1.  Initialize the Image Services Environment.                               */
442  /*      2.  Load the document into the working set.                                  */
443  /*      3.  Print the first (or only) object in the document.                        */
444  /*      4.  Free the working set.                                                    */
445  /*      2.  Return to main() with a return code.                                     */
446  /*                                                                                   */
447  /*ENDPROLOGUE***********************************************************************/
448
449  ULONG PrintDocument ( PRCSTRUCT pRC ) {
450
451      /*****************************************************************************/
452      /*  Define Parameters for SimWsLoadObj                                       */
453      /*****************************************************************************/
454
455      SIMCONTROL_BASE    pControl;          /* Object load control information*/
456      SIMSRCSERVEROBJ    pDataSrc;          /* Source from which to load data */
457      SIMLOAD            pLoadID;           /* Pointer to load handle         */
458
459      /*****************************************************************************/
460      /*  Make sure RCStruct initialized properly                                  */
461      /*****************************************************************************/
462
463      RCStruct.ulRC     = SIM_RC_OK;
464      RCStruct.ulStruct = sizeof(RCSTRUCT);    /* must initialize length field */
465
466      /*****************************************************************************/
467      /* Call SimOpsInitEnv to initialize the environment.                         */
468      /*   - parameter values defined in IP3PRINT.H                                */
469      /*****************************************************************************/
470
471      if (SimOpsInitEnv(PSZAPPLDESC,       /* Application description          */
472                    FALSE,                 /* Is this instance unattended?     */
473                    PSZWORKINGDIR,         /* Working directory for application*/
474                    PSZEXITDIR,            /* User exit directory              */
475                    NULL,
476                    PSZOMANAGER,           /* Folder Mgr Object Manager DLL    */
477                    NULL,
478                    (PRCSTRUCT) &RCStruct )   /* Pointer to RC data structure  */
479      !=  SIM_RC_OK ) {
480          IpsMessage(PSZMSGOPSINITFAIL, RCStruct.ulRC);  /* Display Error Msg  */
481          return( RCStruct.ulRC );
482      }
483
484      /*****************************************************************************/
```

```
    /* Initialize SimWsLoadObj parameters.                                    */
    /*************************************************************************/

    /***************************************/
    /* SIMCONTROL_BASE structure.          */
    /***************************************/

    pControl.ulStruct     = sizeof(SIMCONTROL_BASE);
    pControl.fReserved    = FALSE;
    pControl.pszDesc      = NULL;

    /*********************************/
    /* All attributes are visible.     */
    /*********************************/

    pControl.ulHideFrom   = SIMUSER_NONE;

    /***************************************/
    /* Annotation and mask control - JR     */
    /***************************************/
    pControl.ulDocControl = SIMDOCCTRL_ALLOW_ANNOTATIONS   &
                            SIMDOCCTRL_DISPLAY_ANNOTATIONS &
                            SIMDOCCTRL_PRINT_ANNOTATIONS   &
                            SIMDOCCTRL_ALLOW_MASKS         &
                            SIMDOCCTRL_PRINT_MASKS;

    pControl.ulOptions    = 0;
    pControl.pImport      = NULL;
    pControl.ulImportLen  = 0;

    /***************************************/
    /* SIMSRCSERVEROBJ structure.          */
    /***************************************/

    pDataSrc.hSession     = hSession;
    pDataSrc.Obj          = Obj;
    pDataSrc.ulAccess     = SIM_ACCESS_SHARED_READ;
    pDataSrc.ulPriority   = OM_NORMAL_PRIORITY;

    /*************************************************************************/
    /* Call SimWsLoadObj to load a working set object.                       */
    /*************************************************************************/

    if (SimWsLoadObj(SIMTYPE_BASE,    /* Load object as base document       */
          (USHORT)ulObjConCls,        /* Document content class             */
          SIMREF_NONE,                /* Load as 1st object in working set  */
          (SIMWSOBJ)NULL,             /* Handle to working set object       */
          (PVOID)&pControl,           /* Load control information           */
          SIMSRC_SERVER_OBJ,          /* Document read from object server   */
          (PVOID)&pDataSrc,           /* Server object access description   */
          (PSIMWSOBJ)&hWsObj,         /* Handle of working set object loaded */
          (PSIMLOAD)&pLoadID,         /* Pointer to load handle             */
          (PRCSTRUCT) &RCStruct )     /* Pointer to RC data structure       */
                != SIM_RC_OK ) {
        IpsMessage(PSZMSGWSLOADOBJFAIL, RCStruct.ulRC); /* Display Error Msg */
        return( RCStruct.ulRC );
    }
    /*********************************************************************** */
    /* Code to query POPs                                                    */
    /*********************************************************************  */
//  {
//  ULONG           ulNumberofPOPs = 0;      /* Number of POPs Available */
//  PSIMPROFLIST    pPOPList;                  /* List of POPs */
//  ULONG           ulretcode = 0;
//  USHORT          count;
//  RCSTRUCT        rcStruct;
```

```
551 //
552 //  rcStruct.ulStruct = sizeof(RCSTRUCT);     /* Initialize RCSTRUCT  */
553 //
554 //  /* Is any POP available at this time? */
555 //
556 //  SimPrtQueryPOP(&ulNumberofPOPs, &pPOPList, NULL, &rcStruct);
557 //
558 //  if (rcStruct.ulRC != SIM_RC_OK) {
559 //     IpsMessage("SimPrtQueryPOP error - RC: ", rcStruct.ulRC);
560 //     return( RCStruct.ulRC );
561 //     }
562 //  }
563     /********************************************************************* */
564     /* Code to create a POP                                               */
565     /********************************************************************* */
566 //  {
567 //     ULONG           ulRC = 0;
568 //     PSZ               pszNewPOPDesc;
569 //     RCSTRUCT        rcs;
570 //     rcs.ulStruct = sizeof(RCSTRUCT);
571 //
572 //     printf("Before SimPrtCreatePOP \n");
573 //
574 //     SimPrtCreatePOP("SAMPLEAPPRT",
575 //                              &pszNewPOPDesc,
576 //                              NULLHANDLE,
577 //                              &rcs);
578 //
579 //     if (ulRC != SIM_RC_OK) {
580 //           printf("SimPrtCreatePOP failed %d\n", ulRC);
581 //       }
582 //  }
583     /*********************************************************************/
584     /* Call SimPrtPrintObject to print the object.                       */
585     /*********************************************************************/
586
587     if (SimPrtPrintObject(
588           PSZOS2PRTNAME,              /* Printer Profile (POP) name          */
589           (SIMWSOBJ)hWsObj,           /* Handle of Object to print           */
590           (SIMWIN)NULL,               /* No Display Window Handle provided   */
591           NULL,                       /* No Separator Page                   */
592           0L,                         /* Print control parms (not yet impl)  */
593           (PASYNCCTLSTRUCT)NULL,      /* Null for synchronous operation      */
594           (PRCSTRUCT) &RCStruct )     /* Pointer to RC data structure        */
595 ┌             != SIM_RC_OK ) {
596 │    IpsMessage(PSZMSGPRINTOBJECTFAIL, RCStruct.ulRC); /* Display Err Msg */
597 │    return( RCStruct.ulRC );
598 └}
599 ┌else {
600 │    IpsMessage(PSZMSGPRINTOK, RCStruct.ulRC);
601 └}
602
603     /*********************************************************************/
604     /* Call SimWsDeleteObj to delete the working set.                    */
605     /*********************************************************************/
606
607      SimWsDeleteObj((SIMWSOBJ)hWsObj, /* Handle to working set object       */
608           SIMPURGE_ALL,               /* Delete entire working set           */
609           (PRCSTRUCT) &RCStruct );    /* Pointer to RC data structure        */
610
611     return( RCStruct.ulRC );
612 }
613
```

```
614  /*BEGINPROLOGUE*********************************************************************/
615  /*                                                                                   */
616  /*  Function         : IpsMessage                                                    */
617  /*                                                                                   */
618  /*  Description      : Service function to handle application messages.              */
619  /*                                                                                   */
620  /*  Function type    :  VOID.                                                        */
621  /*                                                                                   */
622  /*  Input Parameters :  (PSZ) Message Text, (ULONG) Error Code                       */
623  /*                                                                                   */
624  /*  Output Parameters:  None.                                                        */
625  /*                                                                                   */
626  /*  Synopsis         :  VOID IpsMessage (PSZ, ULONG)                                 */
627  /*                                                                                   */
628  /*  Return values    :  None.                                                        */
629  /*                                                                                   */
630  /*ENDPROLOGUE***********************************************************************/
631
632  VOID IpsMessage ( PSZ pszMsgText, ULONG ulRC ) {
633
634      char MsgText[80];     /* to format message text */
635      sprintf(MsgText, "%s %d\n", pszMsgText, ulRC);
636
637      /* for PM environment */
638
639      WinMessageBox (HWND_DESKTOP,
640                     HWND_DESKTOP,
641                     MsgText,
642                     "Information",
643                     0,
644                     MB_OK          |
645                     MB_INFORMATION |
646                     MB_APPLMODAL);
647
648      /* for command line environment */
649
650   /* printf(MsgText); */
651
652      return;
653  }
654
655  /*************************************************************************************/
656  /* End of IP3PRINT.C                                                                 */
657  /*************************************************************************************/
```

E.4 IP3PRINT.DEF (10/22/93 08:20:30)

```
1   ;*BEGINPROLOGUE*******************************************************************/
2   ;*                                                                                 */
3   ;* MODULE: IP3PRINT.DEF                                                           */
4   ;*                                                                                 */
5   ;* Description: IP3PRINT.DEF is used by program IP3PRINT.C.                       */
6   ;*                                                                                 */
7   ;* OS          : OS/2 2.1                                                         */
8   ;* Compiler    : C Set/2                                                          */
9   ;*                                                                                 */
10  ;*ENDPROLOGUE*********************************************************************/
11
12  ;*Changed from WINDOWCOMPAT                                                       */
13
14  NAME IP3PRINT WINDOWAPI
15  ;*NAME IP3PRINT WINDOWCOMPAT
```

```
16
17  DESCRIPTION 'ImagePlus Sample Print Program'
18
19  STUB    'OS2STUB.EXE'
20
21  CODE    MOVEABLE
22  DATA    MOVEABLE MULTIPLE
23
24  HEAPSIZE   32768        ; Must be non-zero to use Local memory manager
25  STACKSIZE  32768        ; Must be non-zero for SS == DS
26                          ; suggest 4k as minimum stacksize
27  ;*
28  ;* End of IP3PRINT.DEF
```

E.5 IP3PRINT.MAK (05/03/94 12:02:58)

```
 1  #*****************************************************************************
 2  #
 3  # Make file for IP3PRINT.C                        ImagePlus VisualInfo
 4  #
 5  #*****************************************************************************
 6  #
 7  #  COPYRIGHT:
 8  #  ----------
 9  #  Copyright (C) International Business Machines Corp., 1993, 1994.
10  #
11  #
12  #  DISCLAIMER OF WARRANTIES:
13  #  -------------------------
14  #  The following [enclosed] code is sample code created by IBM
15  #  Corporation.  This sample code is not part of any standard IBM product
16  #  and is provided to you solely for the purpose of assisting you in the
17  #  development of your applications.  The code is provided "AS IS",
18  #  without warranty of any kind.  IBM shall not be liable for any damages
19  #  arising out of your use of the sample code, even if they have been
20  #  advised of the possibility of such damages.
21  #
22  #*****************************************************************************
23  #
24  # External Environment Variables Used
25  #
26  #   SB_ROOT       This environment variable defines where the
27  #                 VisualInfo directory is located.  It maybe set
28  #                 either in the config.sys file or from the
29  #                 command line. For example,
30  #
31  #                     SET SB_ROOT=D:\FRNV1R0
32  #
33  #   DEBUG         This environment variable defines whether a
34  #                 Debug or Non-Debug build is to be performed.
35  #                 It maybe set either in the config.sys file or
36  #                 from the command line(when switching back and
37  #                 forth is desired).  The variable may be set
38  #                 according to
39  #
40  #                     SET DEBUG=1  --> perform Debug build
41  #
42  #                     or,
43  #
44  #                     SET DEBUG=0  --> perform Non-Debug build
45  #
46  #*****************************************************************************
47  #
```

```
48  # Compiler Options Used
49  #
50  #  /C+          Compile only
51  #  /Fd-         Store internal work files in shared memory
52  #  /Gd+         Dynamically link the run-time library
53  #  /Ge+         Build an EXE file
54  #  /Ge-         Build a  DLL file
55  #  /Gh+         Generate code for profiling
56  #  /Gh-         Disable profiling
57  #  /Gm+         Link with multi-threaded version of library
58  #  /I           Include file location
59  #  /Kbcfogapexir  Give all diagnostics except preprocessor trace
60  #  /Lf-         Set all listing options off
61  #  /Mp          Use - optlink - linkage for functions
62  #  /Q+          Do not display logo
63  #  /Re          Generate executable code that can be used in an OS/2 runtime
64  #               environment
65  #  /Rn          Generate executable code that can be used as a subsystem with
66  #               no runtime environment
67  #  /Se          Allow all C Set ++ language extensions except migration
68  #  /Sm          Control Compiler interpretation of unsupported keywords
69  #  /Sn+         Allow use of DBCS
70  #  /Sp1         Align on single byte boundires (i.e pack)
71  #  /Sp4         Align on full word boundires (i.e don't pack)
72  #  /Ss          Allow double slash (//) for comments
73  #  /Ti+         Generate debugging information
74  #
75  #  /ALIGN       Set the alignment factor.  Must be power of 2.
76  #  /CO          Include symbolic debugger info
77  #  /DE          Prepare for debugging.
78  #  /EXEPACK     Compress byte pattern in *.exe
79  #  /MAP         Create *.map file.  Lisat public symbols.
80  #  /NOI         NOIGNORECASE           identifiers are case sensitive
81  #  /NOL         NOLOGO                 disable signon banner
82  #  /NOD         NODEFAULTLIBRARYSEARCH don't search LIB libraries
83  #  /PACKCODE    Pack neighboring code segments.
84  #  /PACKDATA    Pack neighboring data segments.
85  #
86  #*****************************************************************************
87
88  #├"
89  #] Generic/Common Macros                                                    ]
90  #Î┐
91  CCFLAGS  = /C+ /Gd+ /Ge+ /Gm+ /Kb /Mp /Q+ /Re /Se /Sn+ /Sp1 /Ss
92
93  INCLUDES = $(SB_ROOT)\INCLUDE;$(SB_ROOT)\INC
94
95  LLFLAGS  = /NOI /NOL
96
97  #├"
98  #] Set up defaults                                                         ]
99  #Î┐
100
101 !ifndef DEBUG
102 DEBUG=0
103 !endif
104
105 #├"
106 #] Debug/Non-Debug Macros                                                  ]
107 #Î┐
108 !if $(DEBUG)
109 CFLAGS   =  /Gh- /Lf- /Ti+ $(CCFLAGS) /I $(INCLUDES)
110 LFLAGS   = /CO /DE $(LLFLAGS)
111 BINDIR   = .
112 LIBDIR   = $(SB_ROOT)\LIB
113 SRCDIR   = .
```

```
114  OBJDIR   = .
115  SRCFIL   = IP3PRINT
116  !else
117  CFLAGS   = $(CCFLAGS) /I $(INCLUDES)
118  LFLAGS   = $(LLFLAGS)
119  BINDIR   = .
120  LIBDIR   = $(SB_ROOT)\LIB
121  SRCDIR   = .
122  OBJDIR   = .
123  SRCFIL   = IP3PRINT
124  !endif
125
126  COMPILE_REG = ICC $(CFLAGS) -Fo$(OBJDIR)\$@ $(SRCDIR)\$(@B).C
127
128  LINKLIBS = $(LIBDIR)\FRNOFI.LIB    \
129             $(LIBDIR)\FRNOFO.LIB    \
130             $(LIBDIR)\FIWSENV.LIB   \
131             $(LIBDIR)\FIWSP.LIB     \
132             $(LIBDIR)\FIWSWS.LIB    \
133             $(LIBDIR)\FIWSD.LIB     \
134             OS2386.LIB
135
136  OFILES   = $(OBJDIR)\$(SRCFIL).OBJ
137
138  #├"
139  #] Dependencies                                                    ]
140  #Î┐
141
142  FOLDERMGR_DEP = {$(INCLUDES);}FRNP.H         {$(INCLUDES);}FRNPCAPI.H  \
143                  {$(INCLUDES);}FRNPERR.H      {$(INCLUDES);}FRNPFI.H    \
144                  {$(INCLUDES);}FRNPTYPE.H     {$(INCLUDES);}FRNPFI2.H   \
145                  {$(INCLUDES);}FRNPLIBC.H     {$(INCLUDES);}FRNPLCLI.H  \
146                  {$(INCLUDES);}FRNPFO.H
147
148  IS_DEP        = {$(INCLUDES);}FIWS.H         {$(INCLUDES);}FIWSDSP.H   \
149                  {$(INCLUDES);}FIWSWS.H       {$(INCLUDES);}FIWSENV.H   \
150                  {$(INCLUDES);}FIWSPRT.H      {$(INCLUDES);}FIWSDEV.H
151
152  FRNOEXDI_DEP  = {$(INCLUDES);}$(SRCFIL).H     $(FOLDERMGR_DEP)         \
153                   $(IS_DEP)
154
155  #├"
156  #] Main Target Rule                                                ]
157  #Î┐
158  ALL         : ERASE $(BINDIR)\$(SRCFIL).EXE
159
160  ERASE       :
161      if exist ERR.LST erase ERR.LST
162
163  #├"
164  #] Rules                                                           ]
165  #Î┐
166  $(BINDIR)\$(SRCFIL).EXE   :  $(OFILES) $(SRCDIR)\$(SRCFIL).DEF $(SRCFIL).MAK
167     LINK386 $(LFLAGS) $(OFILES),    \
168             $(BINDIR)\$(SRCFIL).EXE, \
169             $(BINDIR)\$(SRCFIL).MAP, \
170             $(LINKLIBS),            \
171             $(SRCDIR)\$(SRCFIL).DEF
172
173  $(OBJDIR)\$(SRCFIL).OBJ : $(SRCDIR)\$(SRCFIL).C $(FRNOEXDI_DEP)
174     $(COMPILE_REG)
```

Appendix F. IP3POP Source Code

F.1 Function index

CreatePOP (IP3POP.C, page 165)	
calls	**IpsMessage** (IP3POP.C, page 167) **SimOpsFreeMem** **SimOpsInitEnv** **SimPrtCreatePOP** **SimPrtQueryPOP**
called by	**main** (IP3POP.C, page 162)

GetDocumentTOC

IpsMessage (IP3POP.C, page 167)	
calls	**sprintf** **WinMessageBox**
called by	**CreatePOP** (IP3POP.C, page 165) **main** (IP3POP.C, page 162)

Logon (IP3POP.C, page 164)	
calls	**SimLibLogon**
called by	**main** (IP3POP.C, page 162)

main (IP3POP.C, page 162)	
calls	**CreatePOP** (IP3POP.C, page 165) **IpsMessage** (IP3POP.C, page 167) **Logon** (IP3POP.C, page 164) **SimLibFree** **SimLibLogoff** **strcpy** **WinCreateMsgQueue** **WinInitialize**

QueryDocument

SimLibFree	
called by	**main** (IP3POP.C, page 162)

SimLibLogoff	
called by	**main** (IP3POP.C, page 162)

SimLibLogon	
called by	**Logon** (IP3POP.C, page 164)

SimOpsFreeMem	
called by	**CreatePOP** (IP3POP.C, page 165)

SimOpsInitEnv	
called by	**CreatePOP** (IP3POP.C, page 165)

SimPrtCreatePOP	
called by	**CreatePOP** (IP3POP.C, page 165)

SimPrtQueryPOP	
called by	**CreatePOP** (IP3POP.C, page 165)

sprintf	
called by	**IpsMessage** (IP3POP.C, page 167)

strcpy	
called by	**main** (IP3POP.C, page 162)

WinCreateMsgQueue	
called by	**main** (IP3POP.C, page 162)

WinInitialize	
called by	**main** (IP3POP.C, page 162)

WinMessageBox	
called by	**IpsMessage** (IP3POP.C, page 167)

F.2 IP3POP.H (05/24/94 14:03:44)

```
1  /*****************************************************************************/
2  /*                                                                           */
3  /* MODULE: IP3POP.H                                                        */
4  /*                                                                           */
5  /* Description  : Header file for module IP3PRINT.C.                         */
6  /*                                                                           */
7  /*****************************************************************************/
8
9  /*****************************************************************************/
10 /* IP3POP.C Function Prototypes                                            */
11 /*****************************************************************************/
12
13 ULONG    Logon              (PRCSTRUCT pRC,
14                             PSZ pszDBName,
15                             PSZ pszApplicationName,
16                             PSZ pszUserID,
17                             PSZ pszPassword) ;
18 ULONG    GetDocumentTOC      (HSESSION hSession,
19                             PRCSTRUCT pRC) ;
20 ULONG    QueryDocument       (HSESSION hSession,
21                             PRCSTRUCT pRC) ;
22 ULONG    CreatePOP           (PRCSTRUCT pRC) ;
23 VOID     IpsMessage          (PSZ   pszMsgText,
24                             ULONG ulRC) ;
25
26 /*****************************************************************************/
27 /* Presentation Manager Definitions                                          */
28 /*****************************************************************************/
29
30 #define INCL_WINWINDOWMGR
31
32 /*****************************************************************************/
33 /* SimLibLogon Parameter Definitions                                         */
34 /*****************************************************************************/
35
36 #define PSZDBNAME          "LIBSRVR2"
37 #define PSZAPPLICATIONNAME "SAMPLE"
38 #define PSZUSERID          "FRNADMIN"
39 #define PSZPASSWORD        "PASSWORD"
40
41 /*****************************************/
42 /* SimLibLogon Parameter Maximum Lengths */
43 /*****************************************/
44
45 #define MAXDBNAME          18          /* Maximum size of pszDBName         */
46 #define MAXAPPLICATIONNAME  8          /* Maximum size of pszApplicationName*/
47 #define MAXUSERID          26          /* Maximum size of pszUserID         */
48 #define MAXPASSWORD         8          /* Maximum size of pszPassword       */
49
50 /*****************************************************************************/
51 /* SimOpsInitEnv Parameter Definitions                                       */
52 /*****************************************************************************/
53
54 #define PSZAPPLDESC        "Sample scenario to print a document."
55 #define PSZWORKINGDIR      "D:\\FRNV1RO\\WORK"
56 /* #define PSZWORKINGDIR    "" /* NULL for Current Dir doesn't seem to work yet */
57 #define PSZEXITDIR         "D:\\FRNV1RO\\DLL"
58 #define PSZOIMANAGER       "FRNOFI"
59 #define PSZOMANAGER        "FRNOFO"
60
```

```
61  /*******************************************/
62  /* SimOpsInitEnv Parameter Maximum Lengths */
63  /*******************************************/
64
65  #define MAXDIRNAME          255        /* Maximum size of pszWorkingDir,    */
66                                         /* pszExitDir, pszOIManager,         */
67                                         /* and pszOManager                   */
68
69  /*************************************************/
70  /* SimOpsInitEnv and SimWsLoadObj Maximum Lengths */
71  /*************************************************/
72
73  #define MAXDESC             40         /* Maximum size of pszApplDesc       */
74                                         /* and pszDocDes                     */
75
76  /*************************************************/
77  /* Printer Option Profile Name for SimPrtPrintObject */
78  /*************************************************/
79
80  #define PSZOS2PRTNAME     "SAMPLEAPPRT"
81
82
83  /*******************************************/
84  /* Message Texts                           */
85  /*******************************************/
86
87  #define PSZMSGPARMERR       "DocID only is required.  Number of parms passed ="
88  #define PSZMSGPRTFAIL       "Print Document Error.  RC ="
89  #define PSZMSGQRYDOCFAIL    "Query Document error.  RC ="
90  #define PSZMSGQRYTOCFAIL    "Query Document TOC error.  RC ="
91  #define PSZMSGLOGONFAIL     "Library Logon error.  RC ="
92  #define PSZMSGNODOCPART     "There are no parts for document "
93  #define PSZMSGOPSINITFAIL   "SimOpsInitEnv error.  RC ="
94  #define PSZMSGWSLOADOBJFAIL "SimWsLoadObj error.  RC ="
95  #define PSZMSGPRINTOBJECTFAIL "SimPrtPrintObject error.  RC ="
96  #define PSZMSGPRINTOK       "Print request completed  RC ="
97
98  /*****************************************************************************/
99  /* End of IP3POP.H                                                           */
100 /*****************************************************************************/
```

F.3 IP3POP.C (06/28/94 14:07:10)

CreatePOP (IP3POP.C, page 165)	
calls	**IpsMessage** (IP3POP.C, page 167) **SimOpsFreeMem** **SimOpsInitEnv** **SimPrtCreatePOP** **SimPrtQueryPOP**
called by	**main** (IP3POP.C, page 162)

IpsMessage (IP3POP.C, page 167)	
calls	**sprintf** **WinMessageBox**
called by	**CreatePOP** (IP3POP.C, page 165) **main** (IP3POP.C, page 162)

Logon (IP3POP.C, page 164)	
calls	**SimLibLogon**
called by	**main** (IP3POP.C, page 162)

main (IP3POP.C, page 162)	
calls	**CreatePOP** (IP3POP.C, page 165) **IpsMessage** (IP3POP.C, page 167) **Logon** (IP3POP.C, page 164) **SimLibFree** **SimLibLogoff** **strcpy** **WinCreateMsgQueue** **WinInitialize**

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/* MODULE: IP3POP.C                                                                 */
/*                                                                                  */
/* DISCLAIMER OF WARRANTIES:                                                        */
/* -------------------------                                                        */
/* The following [enclosed] code is sample code created by IBM                      */
/* Corporation.  This sample code is not part of any standard IBM product           */
/* and is provided to you solely for the purpose of assisting you in the            */
/* development of your applications.  The code is provided "AS IS",                 */
/* without warranty of any kind.  IBM shall not be liable for any damages           */
/* arising out of your use of the sample code, even if they have been               */
/* advised of the possibility of such damages.                                      */
/*                                                                                  */
/* Description : IP3PRINT.c - Sample application code to print a document.          */
/*               Folder Manager APIs used are : SimLibLogon, SimLibFree,            */
/*               SimLibQueryObject and SimLibLogoff.                                */
/*               Image Services APIs used are: SimOpsInitEnv,                       */
/*               SimWsLoadObj, SimPrtPrintObject and SimWsDeleteObj.                */
/*                                                                                  */
/* System      : OS/2 compatible PC                                                 */
/* OS          : OS/2 2.1                                                           */
/* Compiler    : IBM C/C++                                                          */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

/************************************************************************************/
/*  System Header Files                                                             */
/************************************************************************************/
#include <os2.h>                        /* Main OS/2 header files                   */
#include <stdio.h>                      /* Standard I/O header files                */
#include <stdlib.h>                     /* Standard Library header files            */
#include <string.h>                     /* String manipulation                      */

/************************************************************************************/
/*  ImagePlus - Folder Manager header files                                         */
/************************************************************************************/
#define   FRN_INCL_FI                   /* Include frnpfi.h & frnplcli.h            */
#define   FRN_INCL_FO                   /* Include frnpfo.h                         */
#include "frnp.h"                       /* Global types                             */
#include "frnpcapi.h"                   /* Structures and constants                 */

/************************************************************************************/
/*  ImagePlus - Image Services Header Files                                         */
/************************************************************************************/
#define   FIWS_SERVER                   /* Required to read from a server           */
#include "fiws.h"                       /* Base Image Services                      */
#include "fiwsenv.h"                    /* Environment prototypes                   */
#include "fiwsws.h"                     /* Working Set prototypes                   */
#include "fiwsprt.h"                    /* Print prototypes                         */
#include "fiwsdev.h"                    /* Common Device Include - JR               */
/************************************************************************************/
/*  IP3POP Header File                                                            */
/************************************************************************************/
#include "IP3POP.h"                   /* Application header file                 */
/************************************************************************************/
/*  Global Variables                                                                */
/************************************************************************************/
/*  OS/2 and PM APIs                                                                */
/************************************************************************************/
HAB       hab;                          /* Anchor block handle                      */
HMQ       hmq;                          /* Message Queue handle                     */

/************************************************************************************/
/*  ImagePlus APIs - Library Session data used with Folder Manager APIs             */
/************************************************************************************/
```

```
67   HSESSION hSession;                    /* Folder Manager session handle     */
68   RCSTRUCT RCStruct;                    /* RC data structure for API calls   */
69   ULONG    ulMemoryBlock;               /* Generic pointer for SimLibFree    */
70   CHAR     DBName[MAXDBNAME + 1] = "";             /* Library Server Name    */
71   CHAR     ApplicationName[MAXAPPLICATIONNAME +1] = ""; /* Application Name    */
72   CHAR     UserID[MAXUSERID +1] = "";              /* User ID                */
73   CHAR     Password[MAXPASSWORD + 1] = "";         /* Password               */
74   ITEMID   DocItemID;                    /* Document to print                 */
75   BOOL     fPartExists;                  /* Does a part exist for the document*/
76   OBJ      Obj;                          /* Document part to print            */
77   ULONG    ulObjConCls;                  /* Document content class            */
78
79   /*****************************************************************************/
80   /*  ImagePlus APIs - Image Services Data - used with Image Services APIs     */
81   /*****************************************************************************/
82   SIMWSOBJ hWsObj;                       /* Working set object handle         */
83
```

```
84  /*BEGINPROLOGUE*********************************************************************/
85  /*                                                                                  */
86  /*  Function         : main                                                         */
87  /*                                                                                  */
88  /*  Description      : main() function of sample program to print a document        */
89  /*                                                                                  */
90  /*  Function type    :  INT.                                                        */
91  /*                                                                                  */
92  /*  Input Parameters :                                                              */
93  /*                                                                                  */
94  /*  Output Parameters:  None.                                                       */
95  /*                                                                                  */
96  /*  Synopsis         :  INT main ( int argc, char *argv[] )                         */
97  /*                                                                                  */
98  /*  Return values    :  If all functions are successful, the return code            */
99  /*                      from Logoff() is returned.  Otherwise, the return           */
100 /*                      code from the failing function is returned.                 */
101 /*                                                                                  */
102 /*  Process Flow     :                                                              */
103 /*      1.  Initialize Presentation Manager.                                        */
104 /*      2.  Establish a session with the Folder Manager                             */
105 /*      3.  Query Printer Option Profile                                            */
106 /*      4.  Create Printer Option Profile                                           */
107 /*      5.  Logoff from the Folder Manager session.                                 */
108 /*      6.  Exit main() with a return code.                                         */
109 /*                                                                                  */
110 /*ENDPROLOGUE***********************************************************************/
111
112 INT main (int argc, char *argv[]) {
113
114     ULONG ulretcode = 0;
115
116     /*************************************************************************/
117     /* Initialize Session Variables required for logon.                      */
118     /*************************************************************************/
119     strcpy (DBName,            PSZDBNAME);
120     strcpy (ApplicationName,   PSZAPPLICATIONNAME);
121     strcpy (UserID,            PSZUSERID);
122     strcpy (Password,          PSZPASSWORD);
123
124     /*************************************************************************/
125     /* Initialize PM, and create Message Queue so we can use WinMessageBox.  */
126     /* PM is required for Image Services "PRINT" function to operate.        */
127     /*************************************************************************/
128
129     hab = WinInitialize((USHORT) NULL);
130     hmq = WinCreateMsgQueue( hab, 0 );
131     if (hab == NULLHANDLE) {
132         return(99);                                  /* Quit Immediately     */
133     }
134
135     /*************************************************************************/
136     /* Check that Document ID has been passed as parameter.                  */
137     /*************************************************************************/
138
139     /*************************************************************************/
140     /*  Call Logon to establish a Folder Manager Session.                    */
141     /* If successful, call SimLibFree to free the memory for structure       */
142     /* USERLOGONINFOSTRUC returned from SimLibLogon.                         */
143     /*************************************************************************/
144
145     ulretcode = Logon ( &RCStruct,                /* API Return Code struct*/
146                         DBName,                  /* Library Server name   */
147                         ApplicationName,         /* Application Name      */
```

```
148 |                       UserID,                    /* User ID              */
149 |                       Password);                 /* Password             */
150 |
151 |    if (ulretcode == SIM_RC_OK) {
152 |    |
153 |    |    SimLibFree( hSession,                  /* Folder Mgr session handle  */
154 |    |                (PVOID) ulMemoryBlock,     /* Ptr to memory block to free*/
155 |    |                (PRCSTRUCT) &RCStruct );   /* Ptr to RC data structure   */
156 |    }
157 |    else {
158 |    |    IpsMessage(PSZMSGLOGONFAIL, ulretcode);
159 |    |    return(ulretcode);                      /* Quit Immediately     */
160 |    }
161 |
162 |    /*************************************************************************/
163 |    /* Call CreatePOP to create the POP                                      */
164 |    /*************************************************************************/
165 |
166 |    ulretcode = CreatePOP( &RCStruct);
167 |
168 |    if (ulretcode != SIM_RC_OK) {
169 |    |    IpsMessage(PSZMSGPRTFAIL, ulretcode);
170 |    }
171 |
172 |    /*************************************************************************/
173 |    /* Logoff Folder Manager and return.                                     */
174 |    /*************************************************************************/
175 |
176 |    ulretcode = SimLibLogoff( hSession,         /* Folder Mgr session handle */
177 |                  (PASYNCCTLSTRUCT)NULL,        /* Request synch. processing */
178 |                  (PRCSTRUCT) &RCStruct );      /* Ptr to RC data structure  */
179 |    return ( ulretcode );
180 }
181
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : Logon                                                       */
/*                                                                                 */
/*  Description      : This function is called from main() to logon to             */
/*                     ImagePlus Bid Folder Manager.                               */
/*                                                                                 */
/*  Function type    :  ULONG                                                      */
/*                                                                                 */
/*  Input Parameters :  pszDBName pszApplicationName pszUserID pszPassword         */
/*                                                                                 */
/*  Output Parameters:  hSession RCSTRUCT                                          */
/*                                                                                 */
/*  Synopsis         :  ULONG Logon ( pRC                                          */
/*                                    pszDBName,                                   */
/*                                    pszApplicationName,                          */
/*                                    pszUserID,                                   */
/*                                    pszPassword)                                 */
/*                                                                                 */
/*  Return values    :  ulRC return code from SimLibLogon call                     */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*      1.  Initialize SimLibLogon parameters.                                     */
/*      2.  Establish a session with the Folder Manager                            */
/*          by calling SimLibLogon.                                                */
/*      3.  Return to main() with Session Handle                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG Logon (PRCSTRUCT pRC, PSZ pszDBName,
             PSZ pszApplicationName, PSZ pszUserID, PSZ pszPassword) {

    /*************************************************************************/
    /* Initialize SimLibLogon parameters.                                    */
    /*************************************************************************/

    RCStruct.ulRC      = SIM_RC_OK;

    /*************************************************************************/
    /* Call SimLibLogon to establish a Folder Manager session.               */
    /* If successful, save the Folder Manager session handle for subsequent  */
    /* Folder Manager API calls.                                             */
    /*************************************************************************/

    SimLibLogon( (PSZ) DBName,            /* Pointer to Database name        */
                 (PSZ) ApplicationName,   /* Pointer to Application Name     */
                 (PSZ) UserID,            /* Pointer to User Id              */
                 (PSZ) Password,          /* Pointer to Password for User Id */
                 (PSZ) NULL,              /* Logon without proxy             */
                 (PSZ) NULL,              /* Logon without proxy scope       */
                 (PSZ) NULL,
                 SIM_SS_NORMAL,           /* Non-configuration session logon */
                 (PASYNCCTLSTRUCT) NULL,  /* Request synchronous processing  */
                 (PRCSTRUCT) &RCStruct ); /* Pointer to RC data structure    */

    if ( RCStruct.ulRC == SIM_RC_OK) {

        hSession = RCStruct.ulParam1;              /* Save Session Handle   */
                                                   /* Save ptr to USERLOGONINFOSTRUCT */
        ulMemoryBlock = RCStruct.ulParam2;         /* for use with SimLibFree */
    }
    return( RCStruct.ulRC );
}
```

```
247  /*BEGINPROLOGUE*********************************************************/
248  /*                                                                      */
249  /*  Function         : CreatePOP                                        */
250  /*                                                                      */
251  /*  Description      : This function is called from main() to perform the*/
252  /*                     functions necessary to Query and create the POP  */
253  /*                                                                      */
254  /*  Function type    :  ULONG.                                          */
255  /*                                                                      */
256  /*  Input Parameters :  *RCStruct                                       */
257  /*                                                                      */
258  /*  Output Parameters:  None.                                           */
259  /*                                                                      */
260  /*  Synopsis         :  ULONG CreatePOP ( PRCSTRUCT pRC )               */
261  /*                                                                      */
262  /*  Return values    :  If all functions are successful, SIM_RC_OK is   */
263  /*                      returned.  Otherwise the return code from       */
264  /*                      SimOpsInitEnv, SimWsLoadObj, SimPrtPrintObject  */
265  /*                      or SimWsDeleteObj is returned.                  */
266  /*                                                                      */
267  /*  Process Flow     :                                                  */
268  /*      1.  Initialize the Image Services Environment.                  */
269  /*      2.  Query the POP but don't actually do anything with result    */
270  /*      3.  Create  POP                                                 */
271  /*      4.  Free the working set.                                       */
272  /*      5.  Return to main() with a return code.                        */
273  /*                                                                      */
274  /*ENDPROLOGUE***********************************************************/
275
276  ULONG CreatePOP ( PRCSTRUCT pRC ) {
277
278      /************************************************************************/
279      /*  Make sure RCStruct initialized properly                             */
280      /************************************************************************/
281
282      RCStruct.ulRC     = SIM_RC_OK;
283      RCStruct.ulStruct = sizeof(RCSTRUCT);    /* must initialize length field */
284
285      /************************************************************************/
286      /* Call SimOpsInitEnv to initialize the environment.                    */
287      /*   - parameter values defined in IP3PRINT.H                           */
288      /************************************************************************/
289
290    if (SimOpsInitEnv(PSZAPPLDESC,       /* Application description          */
291                  FALSE,                 /* Is this instance unattended?     */
292                  PSZWORKINGDIR,         /* Working directory for application*/
293                  PSZEXITDIR,            /* User exit directory              */
294                  NULL,
295                  PSZOMANAGER,           /* Folder Mgr Object Manager DLL    */
296                  NULL,
297                  (PRCSTRUCT) &RCStruct )   /* Pointer to RC data structure  */
298      {           != SIM_RC_OK ) {
299        IpsMessage(PSZMSGOPSINITFAIL, RCStruct.ulRC);  /* Display Error Msg  */
300        return( RCStruct.ulRC );
301    }
302
303      /************************************************************************/
304      /* Initialize SimWsLoadObj parameters.                                  */
305      /************************************************************************/
306
307  /*********************************************************************** */
308  /* Code to query POPs                                                    */
309  /*********************************************************************  */
310      {
311          ULONG           ulNumberofPOPs = 0;      /* Number of POPs */
312          PSIMPROFLIST    pPOPList;                /* List of POPs */
```

```
313     ULONG           ulRC      = 0;
314     RCSTRUCT        rcStruct;
315
316     rcStruct.ulStruct = sizeof(RCSTRUCT);     /* Initialize RCSTRUCT  */
317     rcStruct.ulRC = 0;
318
319     SimPrtQueryPOP(&ulNumberofPOPs, &pPOPList, NULL, &rcStruct);
320
321     if (rcStruct.ulRC != SIM_RC_OK) {
322     IpsMessage("SimPrtQueryPOP error - RC: ", rcStruct.ulRC);
323     }
324
325     if (rcStruct.ulRC == SIM_RC_OK) {
326     IpsMessage("Number of POPs = ", ulNumberofPOPs);
327     }
328
329     rcStruct.ulStruct = sizeof(RCSTRUCT);     /* Initialize RCSTRUCT  */
330
331     ulRC = SimOpsFreeMem((PVOID)pPOPList, &rcStruct); /* Free Memory  */
332     if (ulRC != 0) {
333     IpsMessage("SimOpsFreeMem error - RC = ", ulRC);
334     } /* endif */
335
336   }
337 /*****************************************************************************/
338 /*  Create POP                                                               */
339 /*****************************************************************************/
340
341   {
342     ULONG           ulRC = 0;
343     PSZ               pszNewPOPDesc;
344     RCSTRUCT        rcs;
345     rcs.ulStruct = sizeof(RCSTRUCT);
346
347        SimPrtCreatePOP("SAMPLEAPPRT",
348                                &pszNewPOPDesc,
349                                NULLHANDLE,
350                                &rcs);
351
352     if (ulRC != SIM_RC_OK) {
353       IpsMessage("SimPrtCreatePOP failed ... RC = ", ulRC);
354       }
355     if (ulRC == SIM_RC_OK) {
356       IpsMessage("SimPrtCreatePOP successful ... RC = ", ulRC);
357       } /* endif */
358   }
359     return( RCStruct.ulRC );
360 }
361
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                  */
/*  Function         : IpsMessage                                                   */
/*                                                                                  */
/*  Description      : Service function to handle application messages.             */
/*                                                                                  */
/*  Function type    :  VOID.                                                       */
/*                                                                                  */
/*  Input Parameters :  (PSZ) Message Text, (ULONG) Error Code                      */
/*                                                                                  */
/*  Output Parameters:  None.                                                       */
/*                                                                                  */
/*  Synopsis         :  VOID IpsMessage (PSZ, ULONG)                                */
/*                                                                                  */
/*  Return values    :  None.                                                       */
/*                                                                                  */
/*ENDPROLOGUE***********************************************************************/

VOID IpsMessage ( PSZ pszMsgText, ULONG ulRC ) {

    char MsgText[80];     /* to format message text */
    sprintf(MsgText, "%s %d\n", pszMsgText, ulRC);

    /* for PM environment */

    WinMessageBox (HWND_DESKTOP,
                   HWND_DESKTOP,
                   MsgText,
                   "Information",
                   0,
                   MB_OK          |
                   MB_INFORMATION |
                   MB_APPLMODAL);

    /* for command line environment */

 /* printf(MsgText); */

    return;
}

/*************************************************************************************/
/* End of IP3POP.C                                                                   */
/*************************************************************************************/
```

F.4 IP3POP.DEF (10/22/93 08:20:30)

```
;*BEGINPROLOGUE*********************************************************************/
;*                                                                                  */
;* MODULE: IP3PRINT.DEF                                                            */
;*                                                                                  */
;* Description: IP3PRINT.DEF is used by program IP3PRINT.C.                        */
;*                                                                                  */
;* OS          : OS/2 2.1                                                          */
;* Compiler    : C Set/2                                                           */
;*                                                                                  */
;*ENDPROLOGUE***********************************************************************/

;*Changed from WINDOWCOMPAT                                                        */

NAME IP3PRINT WINDOWAPI
;*NAME IP3PRINT WINDOWCOMPAT
```

```
16
17  DESCRIPTION 'ImagePlus Sample Print Program'
18
19  STUB    'OS2STUB.EXE'
20
21  CODE    MOVEABLE
22  DATA    MOVEABLE MULTIPLE
23
24  HEAPSIZE   32768        ; Must be non-zero to use Local memory manager
25  STACKSIZE  32768        ; Must be non-zero for SS == DS
26                          ; suggest 4k as minimum stacksize
27  ;*
28  ;* End of IP3PRINT.DEF
```

F.5 IP3POP.MAK (05/12/94 15:38:20)

```
1   #*****************************************************************************
2   #
3   # Make file for IP3POP.C                        ImagePlus VisualInfo
4   #
5   #*****************************************************************************
6   #
7   #  COPYRIGHT:
8   #  ----------
9   #  Copyright (C) International Business Machines Corp., 1993, 1994.
10  #
11  #
12  #  DISCLAIMER OF WARRANTIES:
13  #  -------------------------
14  #  The following [enclosed] code is sample code created by IBM
15  #  Corporation.  This sample code is not part of any standard IBM product
16  #  and is provided to you solely for the purpose of assisting you in the
17  #  development of your applications.  The code is provided "AS IS",
18  #  without warranty of any kind.  IBM shall not be liable for any damages
19  #  arising out of your use of the sample code, even if they have been
20  #  advised of the possibility of such damages.
21  #
22  #*****************************************************************************
23  #
24  # External Environment Variables Used
25  #
26  #   SB_ROOT       This environment variable defines where the
27  #                 VisualInfo directory is located.  It maybe set
28  #                 either in the config.sys file or from the
29  #                 command line. For example,
30  #
31  #                     SET SB_ROOT=D:\FRNV1R0
32  #
33  #   DEBUG         This environment variable defines whether a
34  #                 Debug or Non-Debug build is to be performed.
35  #                 It maybe set either in the config.sys file or
36  #                 from the command line(when switching back and
37  #                 forth is desired).  The variable may be set
38  #                 according to
39  #
40  #                     SET DEBUG=1  --> perform Debug build
41  #
42  #                     or,
43  #
44  #                     SET DEBUG=0  --> perform Non-Debug build
45  #
46  #*****************************************************************************
47  #
```

```
48  # Compiler Options Used
49  #
50  #  /C+          Compile only
51  #  /Fd-         Store internal work files in shared memory
52  #  /Gd+         Dynamically link the run-time library
53  #  /Ge+         Build an EXE file
54  #  /Ge-         Build a  DLL file
55  #  /Gh+         Generate code for profiling
56  #  /Gh-         Disable profiling
57  #  /Gm+         Link with multi-threaded version of library
58  #  /I           Include file location
59  #  /Kbcfogapexir  Give all diagnostics except preprocessor trace
60  #  /Lf-         Set all listing options off
61  #  /Mp          Use - optlink - linkage for functions
62  #  /Q+          Do not display logo
63  #  /Re          Generate executable code that can be used in an OS/2 runtime
64  #               environment
65  #  /Rn          Generate executable code that can be used as a subsystem with
66  #               no runtime environment
67  #  /Se          Allow all C Set ++ language extensions except migration
68  #  /Sm          Control Compiler interpretation of unsupported keywords
69  #  /Sn+         Allow use of DBCS
70  #  /Sp1         Align on single byte boundires (i.e pack)
71  #  /Sp4         Align on full word boundires (i.e don't pack)
72  #  /Ss          Allow double slash (//) for comments
73  #  /Ti+         Generate debugging information
74  #
75  #  /ALIGN       Set the alignment factor.  Must be power of 2.
76  #  /CO          Include symbolic debugger info
77  #  /DE          Prepare for debugging.
78  #  /EXEPACK     Compress byte pattern in *.exe
79  #  /MAP         Create *.map file.  Lisat public symbols.
80  #  /NOI         NOIGNORECASE           identifiers are case sensitive
81  #  /NOL         NOLOGO                 disable signon banner
82  #  /NOD         NODEFAULTLIBRARYSEARCH don't search LIB libraries
83  #  /PACKCODE    Pack neighboring code segments.
84  #  /PACKDATA    Pack neighboring data segments.
85  #
86  #*****************************************************************************
87
88  #├"
89  #] Generic/Common Macros                                                    ]
90  #Î¬
91  CCFLAGS  = /C+ /Gd+ /Ge+ /Gm+ /Kb /Mp /Q+ /Re /Se /Sn+ /Sp1 /Ss
92
93  INCLUDES = $(SB_ROOT)\INCLUDE;$(SB_ROOT)\INC
94
95  LLFLAGS  = /NOI /NOL
96
97  #├"
98  #] Set up defaults                                                         ]
99  #Î¬
100
101 !ifndef DEBUG
102 DEBUG=0
103 !endif
104
105 #├"
106 #] Debug/Non-Debug Macros                                                  ]
107 #Î¬
108 !if $(DEBUG)
109 CFLAGS   =  /Gh- /Lf- /Ti+ $(CCFLAGS) /I $(INCLUDES)
110 LFLAGS   = /CO /DE $(LLFLAGS)
111 BINDIR   = .
112 LIBDIR   = $(SB_ROOT)\LIB
113 SRCDIR   = .
```

```
OBJDIR   = .
SRCFIL   = IP3POP
!else
CFLAGS   = $(CCFLAGS) /I $(INCLUDES)
LFLAGS   = $(LLFLAGS)
BINDIR   = .
LIBDIR   = $(SB_ROOT)\LIB
SRCDIR   = .
OBJDIR   = .
SRCFIL   = IP3POP
!endif

COMPILE_REG = ICC $(CFLAGS) -Fo$(OBJDIR)\$@ $(SRCDIR)\$(@B).C

LINKLIBS = $(LIBDIR)\FRNOFI.LIB    \
           $(LIBDIR)\FRNOFO.LIB    \
           $(LIBDIR)\FIWSENV.LIB   \
           $(LIBDIR)\FIWSP.LIB     \
           $(LIBDIR)\FIWSWS.LIB    \
           $(LIBDIR)\FIWSD.LIB     \
           OS2386.LIB

OFILES   = $(OBJDIR)\$(SRCFIL).OBJ

#┌"
#] Dependencies                                                    ]
#Î┐

FOLDERMGR_DEP = {$(INCLUDES);}FRNP.H         {$(INCLUDES);}FRNPCAPI.H  \
                {$(INCLUDES);}FRNPERR.H      {$(INCLUDES);}FRNPFI.H    \
                {$(INCLUDES);}FRNPTYPE.H     {$(INCLUDES);}FRNPFI2.H   \
                {$(INCLUDES);}FRNPLIBC.H     {$(INCLUDES);}FRNPLCLI.H  \
                {$(INCLUDES);}FRNPFO.H

IS_DEP        = {$(INCLUDES);}FIWS.H         {$(INCLUDES);}FIWSDSP.H   \
                {$(INCLUDES);}FIWSWS.H       {$(INCLUDES);}FIWSENV.H   \
                {$(INCLUDES);}FIWSPRT.H      {$(INCLUDES);}FIWSDEV.H

FRNOEXDI_DEP  = {$(INCLUDES);}$(SRCFIL).H     $(FOLDERMGR_DEP)         \
                 $(IS_DEP)

#┌"
#] Main Target Rule                                                ]
#Î┐
ALL         : ERASE $(BINDIR)\$(SRCFIL).EXE

ERASE       :
    if exist ERR.LST erase ERR.LST

#┌"
#] Rules                                                           ]
#Î┐
$(BINDIR)\$(SRCFIL).EXE   :  $(OFILES) $(SRCDIR)\$(SRCFIL).DEF $(SRCFIL).MAK
   LINK386 $(LFLAGS) $(OFILES),    \
           $(BINDIR)\$(SRCFIL).EXE, \
           $(BINDIR)\$(SRCFIL).MAP, \
           $(LINKLIBS),            \
           $(SRCDIR)\$(SRCFIL).DEF

$(OBJDIR)\$(SRCFIL).OBJ : $(SRCDIR)\$(SRCFIL).C $(FRNOEXDI_DEP)
   $(COMPILE_REG)
```

Appendix G. IP3EXPRT Source Code

G.1 Function index

ExportObject (IP3EXPRT.C, page 183)	
calls	**IpsMessage** (IP3EXPRT.C, page 186) **SimOpsInitEnv** **SimWsDeleteObj** **SimWsLoadObj** **SimWsStoreObj**
called by	**main** (IP3EXPRT.C, page 177)

GetDocumentTOC (IP3EXPRT.C, page 181)	
calls	**IpsMessage** (IP3EXPRT.C, page 186) **SimLibGetItemAffiliatedTOC** **strcat**
called by	**main** (IP3EXPRT.C, page 177)

IpsMessage (IP3EXPRT.C, page 186)	
calls	**printf** **sprintf**
called by	**ExportObject** (IP3EXPRT.C, page 183) **GetDocumentTOC** (IP3EXPRT.C, page 181) **main** (IP3EXPRT.C, page 177)

Logon (IP3EXPRT.C, page 180)	
calls	**SimLibLogon**
called by	**main** (IP3EXPRT.C, page 177)

main (IP3EXPRT.C, page 177)	
calls	**ExportObject** (IP3EXPRT.C, page 183) **GetDocumentTOC** (IP3EXPRT.C, page 181) **IpsMessage** (IP3EXPRT.C, page 186) **Logon** (IP3EXPRT.C, page 180) **QueryDocument** (IP3EXPRT.C, page 182) **SimLibFree** **SimLibLogoff** **strcpy** **strncpy** **strnicmp**

printf	
called by	**IpsMessage** (IP3EXPRT.C, page 186)

QueryDocument (IP3EXPRT.C, page 182)	
calls	**SimLibQueryObject**
called by	**main** (IP3EXPRT.C, page 177)

SimLibFree	
called by	**main** (IP3EXPRT.C, page 177)

SimLibGetItemAffiliatedTOC	
called by	**GetDocumentTOC** (IP3EXPRT.C, page 181)

SimLibLogoff	
called by	**main** (IP3EXPRT.C, page 177)

SimLibLogon	
called by	**Logon** (IP3EXPRT.C, page 180)

SimLibQueryObject	
called by	**QueryDocument** (IP3EXPRT.C, page 182)

SimOpsInitEnv	
called by	**ExportObject** (IP3EXPRT.C, page 183)

SimWsDeleteObj	
called by	**ExportObject** (IP3EXPRT.C, page 183)

SimWsLoadObj	
called by	**ExportObject** (IP3EXPRT.C, page 183)

SimWsStoreObj	
called by	**ExportObject** (IP3EXPRT.C, page 183)

sprintf	
called by	**IpsMessage** (IP3EXPRT.C, page 186)

strncpy	
called by	**main** (IP3EXPRT.C, page 177)

strcat	
called by	**GetDocumentTOC** (IP3EXPRT.C, page 181)

strnicmp	
called by	**main** (IP3EXPRT.C, page 177)

strcpy	
called by	**main** (IP3EXPRT.C, page 177)

G.2 IP3EXPRT.H (05/20/94 15:02:06)

```
1  /*****************************************************************************/
2  /*                                                                           */
3  /* MODULE: IP3EXPRT.H                                                        */
4  /*                                                                           */
5  /* Description  : Header file for module IP3EXPRT.C.                         */
6  /*                                                                           */
7  /*****************************************************************************/
8
9  /*****************************************************************************/
10 /* IP3EXPRT.C Function Prototypes                                            */
11 /*****************************************************************************/
12
13 ULONG    Logon              (PRCSTRUCT pRC,
14                              PSZ pszDBName,
15                              PSZ pszApplicationName,
16                              PSZ pszUserID,
17                              PSZ pszPassword) ;
18 ULONG    GetDocumentTOC      (HSESSION hSession,
19                              PRCSTRUCT pRC) ;
20 ULONG    QueryDocument       (HSESSION hSession,
21                              PRCSTRUCT pRC) ;
22 ULONG    ExportObject        (PRCSTRUCT pRC) ;
23 VOID     IpsMessage          (PSZ   pszMsgText,
24                              ULONG ulRC) ;
25
26 /*****************************************************************************/
27 /* SimLibLogon Parameter Definitions                                         */
28 /*****************************************************************************/
29
30 #define PSZDBNAME          "LIBSRVR2"
31 #define PSZAPPLICATIONNAME "IPSAMP"
32 #define PSZUSERID          "JOHN"
33 #define PSZPASSWORD        "PASSWORD"
34
35 /*****************************************/
36 /* SimLibLogon Parameter Maximum Lengths */
37 /*****************************************/
38
39 #define MAXDBNAME           18          /* Maximum size of pszDBName         */
40 #define MAXAPPLICATIONNAME  8           /* Maximum size of pszApplicationName*/
41 #define MAXUSERID           26          /* Maximum size of pszUserID         */
42 #define MAXPASSWORD         8           /* Maximum size of pszPassword       */
43
```

```
/*****************************************************************************/
/* SimOpsInitEnv Parameter Definitions                                       */
/*****************************************************************************/

#define PSZAPPLDESC         "Sample scenario to Export a document."
#define PSZWORKINGDIR       "D:\\FRNV1RO"
#define PSZEXITDIR          "D:\\FRNV1RO\\DLL"
#define PSZOIMANAGER        "FRNOFI"
#define PSZOMANAGER         "FRNOFO"

/*******************************************/
/* SimOpsInitEnv Parameter Maximum Lengths */
/*******************************************/

#define MAXDIRNAME           255        /* Maximum size of pszWorkingDir,    */
                                        /* pszExitDir, pszOIManager,         */
                                        /* and pszOManager                   */

/*************************************************/
/* SimOpsInitEnv and SimWsLoadObj Maximum Lengths */
/*************************************************/

#define MAXDESC              40         /* Maximum size of pszApplDesc       */
                                        /* and pszDocDes                     */
/***********************************/
/* SimWsStoreObj maximum lengths  */
/***********************************/

#define MAXBUFSIZE      1000000         /* Maximum size of srcBuffer.ulLen   */

/*****************************************/
/* Message Texts                         */
/*****************************************/

#define PSZMSGPARMERR        "Parameter(s) not passed or in error.  Parms passed ="
#define PSZMSGPEXPORTFAIL    "Export Document Error.  RC ="
#define PSZMSGQRYDOCFAIL     "Query Document error.  RC ="
#define PSZMSGQRYTOCFAIL     "Query Document TOC error.  RC ="
#define PSZMSGLOGONFAIL      "Library Logon error.  RC ="
#define PSZMSGNODOCPART      "There are no parts for document "
#define PSZMSGOPSINITFAIL    "SimOpsInitEnv error.  RC ="
#define PSZMSGWSLOADOBJFAIL  "SimWsLoadObj error.  RC ="
#define PSZMSGSTOREOBJECTFAIL "Error Exporting Object.  RC ="
#define PSZMSGWSSTOREOBJFAIL "SimWsStoreObj error.  RC ="
#define PSZMSGFILEOPENFAIL   "Error opening file.  RC ="
#define PSZMSGFILEWRITEFAIL  "Error writing file.  RC ="
#define PSZMSGEXPORTOK       "Export request completed  RC ="

/*****************************************************************************/
/* End of IP3EXPRT.H                                                         */
/*****************************************************************************/
```

G.3 IP3EXPRT.C (06/28/94 14:08:30)

ExportObject (IP3EXPRT.C, page 183)	
calls	**IpsMessage** (IP3EXPRT.C, page 186) **SimOpsInitEnv** **SimWsDeleteObj** **SimWsLoadObj** **SimWsStoreObj**
called by	**main** (IP3EXPRT.C, page 177)

GetDocumentTOC (IP3EXPRT.C, page 181)	
calls	**IpsMessage** (IP3EXPRT.C, page 186) **SimLibGetItemAffiliatedTOC** **strcat**
called by	**main** (IP3EXPRT.C, page 177)

IpsMessage (IP3EXPRT.C, page 186)	
calls	**printf** **sprintf**
called by	**ExportObject** (IP3EXPRT.C, page 183) **GetDocumentTOC** (IP3EXPRT.C, page 181) **main** (IP3EXPRT.C, page 177)

Logon (IP3EXPRT.C, page 180)	
calls	**SimLibLogon**
called by	**main** (IP3EXPRT.C, page 177)

main (IP3EXPRT.C, page 177)	
calls	**ExportObject** (IP3EXPRT.C, page 183) **GetDocumentTOC** (IP3EXPRT.C, page 181) **IpsMessage** (IP3EXPRT.C, page 186) **Logon** (IP3EXPRT.C, page 180) **QueryDocument** (IP3EXPRT.C, page 182) **SimLibFree** **SimLibLogoff** **strcpy** **strncpy** **strnicmp**

QueryDocument (IP3EXPRT.C, page 182)	
calls	**SimLibQueryObject**
called by	**main** (IP3EXPRT.C, page 177)

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/* MODULE: IP3EXPRT.C                                                              */
/*                                                                                 */
/* DISCLAIMER OF WARRANTIES:                                                       */
/* -------------------------                                                       */
/* The following [enclosed] code is sample code created by IBM                     */
/* Corporation.  This sample code is not part of any standard IBM product          */
/* and is provided to you solely for the purpose of assisting you in the           */
/* development of your applications.  The code is provided "AS IS",                */
/* without warranty of any kind.  IBM shall not be liable for any damages         */
/* arising out of your use of the sample code, even if they have been              */
/* advised of the possibility of such damages.                                     */
/*                                                                                 */
/* Description : IP3EXPRT.c - Sample program to export a document part             */
/*                            (object) to a PC file.                               */
/*               Folder Manager APIs used are : SimLibLogon, SimLibFree,           */
/*               SimLibQueryObject and SimLibLogoff.                               */
/*               Image Services APIs used are: SimOpsInitEnv,                      */
/*               SimWsLoadObj, SimWsStoreObj and SimWsDeleteObj.                   */
/*                                                                                 */
/* System      : OS/2 compatible PC                                                */
/* OS          : OS/2 2.1                                                          */
/* Compiler    : IBM C/C++                                                         */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

/*********************************************************************************/
/*  System Header Files                                                          */
/*********************************************************************************/
#include <os2.h>                        /* Main OS/2 header files                */
#include <errno.h>                      /* Standard error handling               */
#include <stdio.h>                      /* Standard I/O header files             */
#include <stdlib.h>                     /* Standard Library header files         */
#include <string.h>                     /* String manipulation                   */

/*********************************************************************************/
/*  ImagePlus - Folder Manager header files                                      */
/*********************************************************************************/
#define   FRN_INCL_FI                   /* Include frnpfi.h & frnplcli.h         */
#define   FRN_INCL_FO                   /* Include frnpfo.h                      */
#include "frnp.h"                       /* Global types                          */
#include "frnpcapi.h"                   /* Structures and constants              */

/*********************************************************************************/
/*  ImagePlus - Image Services Header Files                                      */
/*********************************************************************************/
#define   FIWS_SERVER                   /* Required to read from a server        */
#include "fiws.h"                       /* Base Image Services                   */
#include "fiwsenv.h"                    /* Environment prototypes                */
#include "fiwsws.h"                     /* Working Set prototypes                */

/*********************************************************************************/
/*  IP3EXPRT Header File                                                         */
/*********************************************************************************/
#include "IP3EXPRT.h"                   /* Application header file               */

/*********************************************************************************/
/*  Global Variables                                                             */
/*********************************************************************************/
CHAR     pszFileName[MAXDIRNAME] = "";  /* Object exported to this file          */

/*********************************************************************************/
/*  ImagePlus APIs - Library Session data used with Folder Manager APIs          */
/*********************************************************************************/
HSESSION hSession;                      /* Folder Manager session handle         */
```

```
67  RCSTRUCT RCStruct;                    /* RC data structure for API calls   */
68  ULONG    ulMemoryBlock;               /* Generic pointer for SimLibFree    */
69  CHAR     DBName[MAXDBNAME + 1] = "";              /* Library Server Name    */
70  CHAR     ApplicationName[MAXAPPLICATIONNAME + 1] = ""; /* Application Name   */
71  CHAR     UserID[MAXUSERID + 1] = "";              /* User ID                */
72  CHAR     Password[MAXPASSWORD + 1] = "";          /* Password               */
73  ITEMID   DocItemID;                   /* Document to export                */
74  BOOL     fPartExists;                 /* Does a part exist for the document*/
75  OBJ      Obj;                         /* Document part to export           */
76  ULONG    ulObjConCls;                 /* Document content class            */
77  PVOID    pointr;
78
79  /*****************************************************************************/
80  /*  ImagePlus APIs - Image Services Data - used with Image Services APIs     */
81  /*****************************************************************************/
82  SIMWSOBJ hWsObj;                      /* Working set object handle         */
83
```

```
84  /*BEGINPROLOGUE*********************************************************/
85  /*                                                                      */
86  /*  Function         : main                                             */
87  /*                                                                      */
88  /*  Description      : main() function of sample program to export an object */
89  /*                     to a PC file.                                    */
90  /*                                                                      */
91  /*  Function type    :  INT.                                            */
92  /*                                                                      */
93  /*  Input Parameters :  DocItemID, FileName                             */
94  /*                      e.g. c:>IP3EXPRT /d=W1263629MO511777 /f=test.doc */
95  /*                                                                      */
96  /*  Output Parameters:  None.                                           */
97  /*                                                                      */
98  /*  Synopsis         :  INT main ( int argc, char *argv[] )             */
99  /*                                                                      */
100 /*  Return values    :  If all functions are successful, the return code */
101 /*                      from Logoff() is returned.  Otherwise, the return */
102 /*                      code from the failing function is returned.     */
103 /*                                                                      */
104 /*  Process Flow     :                                                  */
105 /*                                                                      */
106 /*      1.  Establish a session with the Folder Manager                 */
107 /*      2.  If base objects exist for the document:                     */
108 /*          objects exist for the document :                            */
109 /*          - Save the Obj structure of the first document part as      */
110 /*            the document that will be exported.  Free the memory     */
111 /*            containing the array of PAFFTOCENTRYSTRUCT structures     */
112 /*          - Query the object to obtain the document content class     */
113 /*          - Free the memory containing the OBJINFOSTRUCT structure.   */
114 /*      3.  Store the document part from WS to a buffer and write to a file. */
115 /*      4.  Logoff from the Folder Manager session.                     */
116 /*      5.  Exit main() with a return code.                             */
117 /*                                                                      */
118 /*ENDPROLOGUE***********************************************************/
119
120 INT main (int argc, char *argv[]) {
121
122     ULONG ulretcode = 0;                    /* Work variable for return codes */
123     USHORT i;
124     USHORT numParameters         = 0;
125
126     /*************************************************************************/
127     /* Get the file name and DocID from the command line                     */
128     /*************************************************************************/
129
130     for (i=1; i < argc; i++) {
131         if (strnicmp(argv[i],"/D=",3)==0) {
132             strncpy((PSZ) DocItemID, (PSZ) argv[i] + 3, DOC_ID_SIZE);
133             numParameters++;
134         }
135         else if (strnicmp(argv[i],"/F=",3)==0) {
136             strncpy((PSZ) pszFileName, (PSZ) argv[i] + 3, MAXDIRNAME);
137             numParameters++;
138         }
139     }
140
141     if (numParameters != 2) {
142         IpsMessage(PSZMSGPARMERR, (ULONG) numParameters);
143         return(-1);
144     }
145
146     /*************************************************************************/
147     /* Initialize Session Variables required for logon.                      */
148     /*************************************************************************/
```

```
149     strcpy (DBName,           PSZDBNAME);
150     strcpy (ApplicationName,  PSZAPPLICATIONNAME);
151     strcpy (UserID,           PSZUSERID);
152     strcpy (Password,         PSZPASSWORD);
153
154     /*************************************************************************/
155     /*  Call Logon to establish a Folder Manager Session.                    */
156     /* If successful, call SimLibFree to free the memory for structure       */
157     /* USERLOGONINFOSTRUC returned from SimLibLogon.                         */
158     /*************************************************************************/
159     ulretcode = Logon ( &RCStruct,                /* API Return Code struct*/
160                         DBName,                  /* Library Server name   */
161                         ApplicationName,         /* Application Name      */
162                         UserID,                  /* User ID               */
163                         Password);               /* Password              */
164
165     if (ulretcode == SIM_RC_OK) {
166     |
167     |   SimLibFree( hSession,                 /* Folder Mgr session handle  */
168     |               (PVOID) ulMemoryBlock,    /* Ptr to memory block to free*/
169     |               (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure   */
170     }
171     else {
172     |   IpsMessage(PSZMSGLOGONFAIL, ulretcode);
173     |   return(ulretcode);                          /* Quit Immediately      */
174     }
175
176     /*************************************************************************/
177     /*  Call GetDocumentTOC to find the document parts.                      */
178     /*  If successful, and base parts exist for the document, call           */
179     /*  FreeMemory to free the memory for an array of AFFTOCENTRYSTRUCT       */
180     /*  structures returned from SimLibGetItemAffiliatedTOC.                 */
181     /*************************************************************************/
182
183     ulretcode = GetDocumentTOC( hSession, &RCStruct );
184
185     if ( (ulretcode == SIM_RC_OK) && (fPartExists == TRUE) ) {
186     |   SimLibFree( hSession,                 /* Folder Mgr session handle  */
187     |               (PVOID) ulMemoryBlock,    /* Ptr to memory block to free*/
188     |               (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure   */
189     }
190     else {
191     |   IpsMessage(PSZMSGQRYTOCFAIL, ulretcode);
192     |   SimLibLogoff( hSession,                /* Folder Mgr session handle */
193     |                 (PASYNCCTLSTRUCT)NULL,   /* Request synch. processing */
194     |                 (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure  */
195     |   return(ulretcode);                          /* Quit Immediately      */
196     }
197
198     /*************************************************************************/
199     /*  Call QueryDocument to get the content class of the document part.    */
200     /* If successful, call FreeMemory to free the memory for structure       */
201     /* OBJINFOSTRUCT returned from SimLibQueryObject.                        */
202     /*************************************************************************/
203
204     ulretcode = QueryDocument( hSession, &RCStruct );
205
206     if (ulretcode == SIM_RC_OK) {
207     |   SimLibFree( hSession,                 /* Folder Mgr session handle  */
208     |               (PVOID) ulMemoryBlock,    /* Ptr to memory block to free*/
209     |               (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure   */
210     }
211     else {
```

```
212     IpsMessage(PSZMSGQRYDOCFAIL, ulretcode);
213     SimLibLogoff( hSession,                /* Folder Mgr session handle */
214                   (PASYNCCTLSTRUCT)NULL,   /* Request synch. processing */
215                   (PRCSTRUCT) &RCStruct );  /* Ptr to RC data structure  */
216     return(ulretcode);                       /* Quit Immediately      */
217   }
218
219   /*************************************************************************/
220   /* Call ExportObject to copy the document part to a PC file.             */
221   /*************************************************************************/
222
223   ulretcode = ExportObject( &RCStruct);
224
225   if (ulretcode != SIM_RC_OK) {
226     IpsMessage(PSZMSGSTOREOBJECTFAIL, ulretcode);
227   }
228
229   /*************************************************************************/
230   /* Logoff Folder Manager and return.                                     */
231   /*************************************************************************/
232
233   ulretcode = SimLibLogoff( hSession,         /* Folder Mgr session handle */
234                 (PASYNCCTLSTRUCT)NULL,        /* Request synch. processing */
235                 (PRCSTRUCT) &RCStruct );      /* Ptr to RC data structure  */
236   return ( ulretcode );
237 }
238
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : Logon                                                       */
/*                                                                                 */
/*  Description      : This function is called from main() to logon to             */
/*                     ImagePlus Bid Folder Manager.                               */
/*                                                                                 */
/*  Function type    :  ULONG                                                      */
/*                                                                                 */
/*  Input Parameters :  pszDBName pszApplicationName pszUserID pszPassword         */
/*                                                                                 */
/*  Output Parameters:  hSession RCSTRUCT                                          */
/*                                                                                 */
/*  Synopsis         :  ULONG Logon ( pRC                                          */
/*                                    pszDBName,                                   */
/*                                    pszApplicationName,                          */
/*                                    pszUserID,                                   */
/*                                    pszPassword)                                 */
/*                                                                                 */
/*  Return values    :  ulRC return code from SimLibLogon call                     */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*                                                                                 */
/*      1.  Initialize SimLibLogon parameters.                                     */
/*      2.  Establish a session with the Folder Manager                            */
/*          by calling SimLibLogon.                                                */
/*      3.  Return to main() with Session Handle                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG Logon (PRCSTRUCT pRC, PSZ pszDBName,
             PSZ pszApplicationName, PSZ pszUserID, PSZ pszPassword) {

    /*********************************************************************************/
    /* Initialize SimLibLogon parameters.                                            */
    /*********************************************************************************/

    RCStruct.ulRC      = SIM_RC_OK;

    /*********************************************************************************/
    /* Call SimLibLogon to establish a Folder Manager session.                       */
    /* If successful, save the Folder Manager session handle for subsequent          */
    /* Folder Manager API calls.                                                     */
    /*********************************************************************************/

    SimLibLogon( (PSZ) DBName,            /* Pointer to Database name          */
                 (PSZ) ApplicationName,   /* Pointer to Application Name       */
                 (PSZ) UserID,            /* Pointer to User Id                */
                 (PSZ) Password,          /* Pointer to Password for User Id   */
                 (PSZ) NULL,              /* New Password                      */
                 (PSZ) NULL,              /* Logon without proxy               */
                 (PSZ) NULL,              /* Logon without proxy scope         */
                 SIM_SS_NORMAL,           /* Non-configuration session logon   */
                 (PASYNCCTLSTRUCT) NULL,  /* Request synchronous processing    */
                 (PRCSTRUCT) &RCStruct ); /* Pointer to RC data structure      */

    if ( RCStruct.ulRC == SIM_RC_OK) {

        hSession = RCStruct.ulParam1;               /* Save Session Handle     */
                                                    /* Save ptr to USERLOGONINFOSTRUCT */
        ulMemoryBlock = RCStruct.ulParam2;          /* for use with SimLibFree */
    }
    return( RCStruct.ulRC );
}
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : GetDocumentTOC                                              */
/*                                                                                 */
/*  Description      : This function is called from main() to retrieve             */
/*                     the table of contents for a document.  The first            */
/*                     document part will be exported.                             */
/*                                                                                 */
/*  Function type    :  ULONG.                                                     */
/*                                                                                 */
/*  Input Parameters :  hSession, *RCStruct                                        */
/*                                                                                 */
/*  Output Parameters:  None.                                                      */
/*                                                                                 */
/*  Synopsis         :  ULONG GetDocumentTOC ( hSession, *RCStruct ).              */
/*                                                                                 */
/*  Return values    :  The return code from SimLibGetItemAffiliatedTOC.           */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*                                                                                 */
/*      1.  Call the SimLibGetItemAffiliatedTOC Folder Manager API.                */
/*      2.  Extract and save object data of first object in document.              */
/*      3.  Return to main() with a return code.                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG GetDocumentTOC ( HSESSION hSession, PRCSTRUCT pRC ) {

    PAFFTOCENTRYSTRUCT DocuTOC;            /* TOC data structure            */
    USHORT             count;              /* Loop counter                  */

    /*************************************************************************/
    /* Call SimLibGetItemAffiliatedTOC to get the table of contents for      */
    /* a document.                                                           */
    /* If parts exist for the document, save the pointer to the array of     */
    /* AFFTOCENTRYSTRUCT structures to allow memory to be freed later with   */
    /* SimLibFree.                                                           */
    /*************************************************************************/

    SimLibGetItemAffiliatedTOC(hSession, /* Folder Manager session handle    */
         (PITEMID) DocItemID,           /* Get TOC for this Item ID         */
         SIM_BASE,                      /* Define object type as base       */
         (PASYNCCTLSTRUCT) NULL,        /* Request synchronous processing   */
         (PRCSTRUCT) &RCStruct );       /* Pointer to RC data structure     */

    if (RCStruct.ulRC == SIM_RC_OK) {
        if ((PAFFTOCENTRYSTRUCT)RCStruct.ulParam2 == (PAFFTOCENTRYSTRUCT)NULL)
        {
            fPartExists = FALSE;
            IpsMessage(strcat(PSZMSGLOGONFAIL, (PSZ)DocItemID), RCStruct.ulRC);
        }
        else {
            ulMemoryBlock = RCStruct.ulParam1;
            fPartExists = TRUE;
            DocuTOC = (PAFFTOCENTRYSTRUCT)RCStruct.ulParam1;

    /*************************************************************************/
    /* Save the Obj structure of the first part associated with the          */
    /* document as the part that will be exported.                           */
    /*************************************************************************/
            count = 0;                      /* First Object (Document Part) */
            Obj = DocuTOC[count].Obj;       /* save what we need            */
        }
    }

    return( RCStruct.ulRC );
```

```
370 }
371
372 /*BEGINPROLOGUE*********************************************************************/
373 /*                                                                                  */
374 /*  Function         : QueryDocument                                                */
375 /*                                                                                  */
376 /*  Description      : This function is called from main() to determine the         */
377 /*                     content class of the document to be exported.                */
378 /*                                                                                  */
379 /*  Function type    :  ULONG.                                                      */
380 /*                                                                                  */
381 /*  Input Parameters :  hSession, *RCStruct                                         */
382 /*                                                                                  */
383 /*  Output Parameters:  None.                                                       */
384 /*                                                                                  */
385 /*  Synopsis         :  ULONG QueryDocument  ( hSession, *RCStruct ).               */
386 /*                                                                                  */
387 /*  Return values    :  Return code from SimLibQueryObject.                         */
388 /*                                                                                  */
389 /*  Process Flow     :                                                              */
390 /*                                                                                  */
391 /*      1.  Call SimLibQueryObject Folder Manager API using the Obj                 */
392 /*          structure returned from SimLibGetItemAffiliatedTOC.                     */
393 /*      2.  Return to main() with a return code.                                    */
394 /*                                                                                  */
395 /*ENDPROLOGUE***********************************************************************/
396
397 ULONG QueryDocument ( HSESSION hSession, PRCSTRUCT pRC ) {
398
399     POBJINFOSTRUCT ObjInfo;             /* Object information structure      */
400
401     /*************************************************************************/
402     /* Call SimLibQueryObject to find the content class of the object.       */
403     /* Save the pointer to the OBJINFOSTRUCT structure so that the memory    */
404     /* can be freed later using SimLibFree.                                  */
405     /*************************************************************************/
406
407     SimLibQueryObject(hSession,              /* Folder Mgr session handle   */
408                       (HOBJ) &Obj,           /* ItemID of object to query   */
409                       (PASYNCCTLSTRUCT)NULL, /* Request synch. processing   */
410                       (PRCSTRUCT) &RCStruct );/* Ptr to RC data structure   */
411
412     if (RCStruct.ulRC == SIM_RC_OK) {
413         ulMemoryBlock = RCStruct.ulParam1;  /* Save ptr for later SimLibFree */
414         ObjInfo       =  (POBJINFOSTRUCT)RCStruct.ulParam1;
415         ulObjConCls   =   ObjInfo->ulObjConCls;
416     }
417
418     return( RCStruct.ulRC );           /* Return code from SimLibQueryObject */
419 }
420
```

```
/*BEGINPROLOGUE*********************************************************************/
/*                                                                                 */
/*  Function         : ExportObject                                                */
/*                                                                                 */
/*  Description      : This function is called from main() to perform the          */
/*                     functions necessary to export the first object in           */
/*                     a document.                                                 */
/*                                                                                 */
/*  Function type    :  ULONG.                                                     */
/*                                                                                 */
/*  Input Parameters :  *RCStruct                                                  */
/*                                                                                 */
/*  Output Parameters:  None.                                                      */
/*                                                                                 */
/*  Synopsis         :  ULONG ExportObject( PRCSTRUCT pRC )                        */
/*                                                                                 */
/*  Return values    :  If all functions are successful, SIM_RC_OK is              */
/*                      returned.  Otherwise the return code from                  */
/*                      SimOpsInitEnv, SimWsLoadObj, SimWsStoreObj                 */
/*                      or SimWsDeleteObj is returned.                             */
/*                                                                                 */
/*  Process Flow     :                                                             */
/*                                                                                 */
/*      1.  Initialize the Image Services Environment.                             */
/*      2.  Load the first object in the document into the working set.            */
/*      3.  Write the object fron the working set to a PC file.                    */
/*      4.  Free the working set.                                                  */
/*      2.  Return to main() with a return code.                                   */
/*                                                                                 */
/*ENDPROLOGUE***********************************************************************/

ULONG ExportObject ( PRCSTRUCT pRC ) {

    /*************************************************************************/
    /*  Define Parameters for SimWsLoadObj and SimWsStoreObj                 */
    /*************************************************************************/

    SIMCONTROL_BASE    pControl;          /* Object load control information*/
    SIMSRCSERVEROBJ    pDataSrc;          /* Source from which to load data */
    SIMLOAD            pLoadID;           /* Pointer to load handle         */

    /*************************************************************************/
    /*  Make sure RCStruct initialized properly                              */
    /*************************************************************************/

    RCStruct.ulStruct = sizeof(RCSTRUCT);    /* must initialize length field */

    /*************************************************************************/
    /* Call SimOpsInitEnv to initialize the environment.                     */
    /*   - parameter values defined in IP3EXPRT.H                            */
    /*************************************************************************/

    if (SimOpsInitEnv(PSZAPPLDESC,       /* Application description          */
                  FALSE,                 /* Is this instance unattended?     */
                  PSZWORKINGDIR,         /* Working directory for application*/
                  PSZEXITDIR,            /* User exit directory              */
                  NULL,
                  PSZOMANAGER,          /* Folder Manager object     Mgr DLL */
                  NULL,
                  (PRCSTRUCT) &RCStruct )   /* Pointer to RC data structure  */
                  != SIM_RC_OK ) {
        IpsMessage(PSZMSGOPSINITFAIL, RCStruct.ulRC);  /* Display Error Msg  */
        return( RCStruct.ulRC );
    }

    /*************************************************************************/
```

```
487     /* Initialize SimWsLoadObj parameters.                                  */
488     /*************************************************************************/
489
490     /***************************************/
491     /* SIMCONTROL_BASE structure.          */
492     /***************************************/
493
494     pControl.ulStruct     = sizeof(SIMCONTROL_BASE);
495     pControl.fReserved    = FALSE;
496     pControl.pszDesc      = NULL;
497     pControl.pImport      = NULL;
498     pControl.ulImportLen     = 0;
499
500     /***********************************/
501     /* All attributes are visible.     */
502     /***********************************/
503
504     pControl.ulHideFrom   = SIMUSER_NONE;
505
506     /***************************************/
507     /* Annotation and mask control - JR    */
508     /***************************************/
509     pControl.ulDocControl = SIMDOCCTRL_ALLOW_ANNOTATIONS   &
510                             SIMDOCCTRL_DISPLAY_ANNOTATIONS &
511                             SIMDOCCTRL_PRINT_ANNOTATIONS   &
512                             SIMDOCCTRL_ALLOW_MASKS         &
513                             SIMDOCCTRL_PRINT_MASKS;
514     pControl.ulOptions = 0;
515     /***************************************/
516     /* SIMSRCSERVEROBJ structure.          */
517     /***************************************/
518
519     pDataSrc.hSession     = hSession;
520     pDataSrc.Obj          = Obj;
521     pDataSrc.ulAccess     = SIM_ACCESS_SHARED_READ;
522     pDataSrc.ulPriority   = OM_NORMAL_PRIORITY;
523
524     /*************************************************************************/
525     /* Call SimWsLoadObj to load a working set object.                       */
526     /*************************************************************************/
527
528     if (SimWsLoadObj(SIMTYPE_BASE,    /* Load object as base document       */
529           (USHORT)ulObjConCls,        /* Document content class             */
530           SIMREF_NONE,                /* Load as 1st object in working set  */
531           (SIMWSOBJ)NULL,             /* Handle to working set object       */
532           (PVOID)&pControl,           /* Load control information           */
533           SIMSRC_SERVER_OBJ,          /* Document read from object server   */
534           (PVOID)&pDataSrc,           /* Server object access description   */
535           (PSIMWSOBJ)&hWsObj,         /* Handle of working set object loaded */
536           (PSIMLOAD)&pLoadID,         /* Pointer to load handle             */
537           /*(PASYNCCTLSTRUCT)NULL,      Request synchronous processing     */
538           (PRCSTRUCT) &RCStruct )     /* Pointer to RC data structure       */
539 ┌             != SIM_RC_OK ) {
540 │     IpsMessage(PSZMSGWSLOADOBJFAIL, RCStruct.ulRC); /* Display Error Msg */
541 │     return( RCStruct.ulRC );
542 └─}
543
544     /*************************************************************/
545     /* Call SimWsStoreObj to STORE the object into a pcfile.     */
546     /*************************************************************/
547
548     /*************************************************************/
549     /* Initialize SimWsStoreObj parameters.                      */
550     /*************************************************************/
551
552     RCStruct.ulStruct = sizeof(RCSTRUCT);    /* must initialize length field */
```

```
553
554     /* SimWsStoreObj requires filename in a pointer to a pointer */
555         pointr = &pszFileName ;
556
557         SimWsStoreObj(
558                hWsObj,
559                ulObjConCls,
560                TRUE,
561                SIMSRC_FILENAME,
562                &pointr,
563                  NULL,
564                (PRCSTRUCT) &RCStruct);
565
566        if (RCStruct.ulRC != SIM_RC_OK) {
567          IpsMessage(PSZMSGWSSTOREOBJFAIL, RCStruct.ulRC); /* Display Error Msg */
568          return( RCStruct.ulRC );
569        }
570
571
572     /*************************************************************************/
573     /* Call SimWsDeleteObj to delete the working set.                        */
574     /*************************************************************************/
575
576      SimWsDeleteObj((SIMWSOBJ)hWsObj, /* Handle to working set object        */
577           SIMPURGE_ALL,               /* Delete entire working set           */
578           (PRCSTRUCT) &RCStruct );    /* Pointer to RC data structure        */
579
580     return( RCStruct.ulRC );
581 }
582
```

```
583  /*BEGINPROLOGUE*********************************************************************/
584  /*                                                                                  */
585  /*  Function         : IpsMessage                                                   */
586  /*                                                                                  */
587  /*  Description      : Service function to handle application messages.             */
588  /*                                                                                  */
589  /*  Function type    :  VOID.                                                       */
590  /*                                                                                  */
591  /*  Input Parameters :  (PSZ) Message Text, (ULONG) Error Code                      */
592  /*                                                                                  */
593  /*  Output Parameters:  None.                                                       */
594  /*                                                                                  */
595  /*  Synopsis         :  VOID IpsMessage (PSZ, ULONG)                                */
596  /*                                                                                  */
597  /*  Return values    :  None.                                                       */
598  /*                                                                                  */
599  /*ENDPROLOGUE***********************************************************************/
600
601  VOID IpsMessage ( PSZ pszMsgText, ULONG ulRC ) {
602
603      char MsgText[80];     /* to format message text */
604      sprintf(MsgText, "%s %d\n", pszMsgText, ulRC);
605
606      /* for PM environment */
607
608      /* WinMessageBox (HWND_DESKTOP,
609                     HWND_DESKTOP,
610                     MsgText,
611                     "Information",
612                     0,
613                     MB_OK          |
614                     MB_INFORMATION |
615                     MB_APPLMODAL); */
616
617      /* for command line environment */
618
619      printf(MsgText);
620
621      return;
622  }
623
624  /*************************************************************************************/
625  /* End of IP3EXPRT.C                                                                 */
626  /*************************************************************************************/
```

G.4 IP3EXPRT.DEF (10/25/93 06:52:20)

```
1   ;*BEGINPROLOGUE*********************************************************************/
2   ;*                                                                                  */
3   ;* MODULE: IP3EXPRT.DEF                                                           */
4   ;*                                                                                  */
5   ;* Description: IP3EXPRT.DEF is used by program IP3EXPRT.C.                       */
6   ;*                                                                                  */
7   ;*ENDPROLOGUE***********************************************************************/
8
9
10  NAME IP3EXPRT WINDOWCOMPAT
11
12  DESCRIPTION 'Spectacular Bid Sample Application'
13
14  STUB    'OS2STUB.EXE'
15
```

```
16  CODE    MOVEABLE
17  DATA    MOVEABLE MULTIPLE
18
19  HEAPSIZE   65536        ; Must be non-zero to use Local memory manager
20  STACKSIZE  65536        ; Must be non-zero for SS == DS
21                          ; suggest 4k as minimum stacksize
22  ;*
23  ;* End of IP3EXPRT.DEF
```

G.5 IP3EXPRT.MAK (05/20/94 15:02:42)

```
#*****************************************************************************
#
# Make file for IP3EXPRT.C                        ImagePlus VisualInfo
#
#*****************************************************************************
#
#  COPYRIGHT:
#  ----------
#  Copyright (C) International Business Machines Corp., 1993, 1994.
#
#
#  DISCLAIMER OF WARRANTIES:
#  -------------------------
#  The following [enclosed] code is sample code created by IBM
#  Corporation.  This sample code is not part of any standard IBM product
#  and is provided to you solely for the purpose of assisting you in the
#  development of your applications.  The code is provided "AS IS",
#  without warranty of any kind.  IBM shall not be liable for any damages
#  arising out of your use of the sample code, even if they have been
#  advised of the possibility of such damages.
#
#*****************************************************************************
#
# External Environment Variables Used
#
#   SB_ROOT       This environment variable defines where the
#                 VisualInfo directory is located.  It maybe set
#                 either in the config.sys file or from the
#                 command line. For example,
#
#                     SET SB_ROOT=D:\FRNV1R0
#
#   DEBUG         This environment variable defines whether a
#                 Debug or Non-Debug build is to be performed.
#                 It maybe set either in the config.sys file or
#                 from the command line(when switching back and
#                 forth is desired).  The variable may be set
#                 according to
#
#                     SET DEBUG=1  --> perform Debug build
#
#                     or,
#
#                     SET DEBUG=0  --> perform Non-Debug build
#
#*****************************************************************************
#
# Compiler Options Used
#
#  /C+          Compile only
#  /Fd-         Store internal work files in shared memory
#  /Gd+         Dynamically link the run-time library
```

```
53  #  /Ge+         Build an EXE file
54  #  /Ge-         Build a  DLL file
55  #  /Gh+         Generate code for profiling
56  #  /Gh-         Disable profiling
57  #  /Gm+         Link with multi-threaded version of library
58  #  /I           Include file location
59  #  /Kbcfogapexir  Give all diagnostics except preprocessor trace
60  #  /Lf-         Set all listing options off
61  #  /Mp          Use - optlink - linkage for functions
62  #  /Q+          Do not display logo
63  #  /Re          Generate executable code that can be used in an OS/2 runtime
64  #               environment
65  #  /Rn          Generate executable code that can be used as a subsystem with
66  #               no runtime environment
67  #  /Se          Allow all C Set ++ language extensions except migration
68  #  /Sm          Control Compiler interpretation of unsupported keywords
69  #  /Sn+         Allow use of DBCS
70  #  /Sp1         Align on single byte boundires (i.e pack)
71  #  /Sp4         Align on full word boundires (i.e don't pack)
72  #  /Ss          Allow double slash (//) for comments
73  #  /Ti+         Generate debugging information
74  #
75  #  /ALIGN       Set the alignment factor.  Must be power of 2.
76  #  /CO          Include symbolic debugger info
77  #  /DE          Prepare for debugging.
78  #  /EXEPACK     Compress byte pattern in *.exe
79  #  /MAP         Create *.map file.  Lisat public symbols.
80  #  /NOI         NOIGNORECASE           identifiers are case sensitive
81  #  /NOL         NOLOGO                 disable signon banner
82  #  /NOD         NODEFAULTLIBRARYSEARCH don't search LIB libraries
83  #  /PACKCODE    Pack neighboring code segments.
84  #  /PACKDATA    Pack neighboring data segments.
85  #
86  #*****************************************************************************
87
88  #├"
89  #] Generic/Common Macros                                                    ]
90  #Î┐
91  CCFLAGS  = /C+ /Gd+ /Ge+ /Gm+ /Kb /Mp /Q+ /Re /Se /Sn+ /Sp1 /Ss
92
93  INCLUDES = $(SB_ROOT)\INCLUDE;$(SB_ROOT)\INC
94
95  LLFLAGS  = /NOI /NOL
96
97  #├"
98  #] Set up defaults                                                         ]
99  #Î┐
100
101 !ifndef DEBUG
102 DEBUG=0
103 !endif
104
105 #├"
106 #] Debug/Non-Debug Macros                                                  ]
107 #Î┐
108 !if $(DEBUG)
109 CFLAGS   =  /Gh- /Lf- /Ti+ $(CCFLAGS) /I $(INCLUDES)
110 LFLAGS   = /CO /DE $(LLFLAGS)
111 BINDIR   = .
112 LIBDIR   = $(SB_ROOT)\LIB
113 SRCDIR   = .
114 OBJDIR   = .
115 SRCFIL   = IP3EXPRT
116 !else
117 CFLAGS   = $(CCFLAGS) /I $(INCLUDES)
118 LFLAGS   = $(LLFLAGS)
```

```
BINDIR   = .
LIBDIR   = $(SB_ROOT)\LIB
SRCDIR   = .
OBJDIR   = .
SRCFIL   = IP3EXPRT
!endif

COMPILE_REG = ICC $(CFLAGS) -Fo$(OBJDIR)\$@ $(SRCDIR)\$(@B).C

LINKLIBS = $(LIBDIR)\FRNOFI.LIB    \
           $(LIBDIR)\FRNOFO.LIB    \
           $(LIBDIR)\FIWSENV.LIB   \
           $(LIBDIR)\FIWSWS.LIB    \
           $(LIBDIR)\FIWSD.LIB     \
           OS2386.LIB

OFILES   = $(OBJDIR)\$(SRCFIL).OBJ

#├"
#] Dependencies                                                    ]
#Î┐

FOLDERMGR_DEP = {$(INCLUDES);}FRNP.H         {$(INCLUDES);}FRNPCAPI.H  \
                {$(INCLUDES);}FRNPERR.H      {$(INCLUDES);}FRNPFI.H    \
                {$(INCLUDES);}FRNPTYPE.H     {$(INCLUDES);}FRNPFI2.H   \
                {$(INCLUDES);}FRNPLIBC.H     {$(INCLUDES);}FRNPLCLI.H  \
                {$(INCLUDES);}FRNPFO.H

IS_DEP        = {$(INCLUDES);}FIWS.H         {$(INCLUDES);}FIWSDSP.H   \
                {$(INCLUDES);}FIWSWS.H       {$(INCLUDES);}FIWSENV.H

FRNOEXDI_DEP  = {$(INCLUDES);}$(SRCFIL).H     $(FOLDERMGR_DEP)          \
                 $(IS_DEP)

#├"
#] Main Target Rule                                                ]
#Î┐
ALL         : ERASE $(BINDIR)\$(SRCFIL).EXE

ERASE       :
    if exist ERR.LST erase ERR.LST

#├"
#] Rules                                                           ]
#Î┐
$(BINDIR)\$(SRCFIL).EXE   :  $(OFILES) $(SRCDIR)\$(SRCFIL).DEF $(SRCFIL).MAK
   LINK386 $(LFLAGS) $(OFILES),    \
           $(BINDIR)\$(SRCFIL).EXE, \
           $(BINDIR)\$(SRCFIL).MAP, \
           $(LINKLIBS),            \
           $(SRCDIR)\$(SRCFIL).DEF

$(OBJDIR)\$(SRCFIL).OBJ : $(SRCDIR)\$(SRCFIL).C $(FRNOEXDI_DEP)
   $(COMPILE_REG)
```

Index

Special Characters

Numerics

A

B

C

D

E

F

H

I

L

M

N

O

P

Q

R

S

ITSO Technical Bulletin Evaluation RED000

ImagePlus VisualInfo
Client/Server Solution:
Sample Code for Client

Publication No. GG24-4369-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction	____		
Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:
   Do you provide billable services for 20% or more of your time? Yes____ No____
   Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____
   If no, please explain:

_____

_____

What other topics would you like to see in this Bulletin?

_____

_____

What other Technical Bulletins would you like to see published?

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____ _____
Name Address

_____ _____
Company or Organization

_____ _____
Phone No.

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department FGC, Location BME
6710 Rockledge Drive
Bethesda, MD
USA 20817-1824

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.