AS/400 Advanced Series

Finance Communications Programming

Version 4

SC41-5449-00

AS/400 Advanced Series

Finance Communications Programming

Version 4

SC41-5449-00

Take Note!

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

First Edition (August 1997)

This edition applies to the licensed program IBM Operating System/400 (Program 5769-SS1), Version 4 Release 1 Modification 0, and to all subsequent releases and modifications until otherwise indicated in new editions.

Make sure that you are using the proper edition for the level of the product.

Order publications through your IBM representative or the IBM branch serving your locality. If you live in the United States, Puerto Rico, or Guam, you can order publications through the IBM Software Manufacturing Solutions at 800+879-2755. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for readers’ comments may be provided at the back of this publication. You can also mail your comments to the following address:

IBM Corporation
Attention Department 542
IDCLERK
3605 Highway 52 N
Rochester, MN 55901-7829 USA

or you can fax your comments to:

United States and Canada: 800+937-3430
Other countries: (+1)+507+253-5192

If you have access to Internet, you can send your comments electronically to IDCLERK@RCHVMW2.VNET.IBM.COM; IBMMAIL, to IBMMAIL(USIB56RZ).

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

Contents

Figures

Notices

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 that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact the software interoperability coordinator. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Address your questions to:

IBM Corporation
Software Interoperability Coordinator
3605 Highway 52 N
Rochester, MN 55901-7829 USA

This publication could contain technical inaccuracies or typographical errors.

This publication may refer to products that are announced but not currently available in your country. This publication may also refer to products that have not been announced in your country. IBM makes no commitment to make available any unannounced products referred to herein. The final decision to announce any product is based on IBM's business and technical judgment.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This publication contains small programs that are furnished by IBM as simple examples to provide an illustration. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. All programs contained herein are provided to you "AS IS". THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Programming Interface Information

This book is intended to help the customer to use the finance communications support. This book documents General-Use Programming Interface and Associated Guidance Information provided by OS/400 program.

General-Use programming interfaces allow the customer to write programs that obtain the services of OS/400 program.

The information in Chapter 6, “Writing Non-Intersystem Communications Function Application Programs” is intended to help the customer write finance application programs using the non-ICF interface. It contains general-use programming interfaces, which allow the customer to write programs that use the services of the OS/400 program.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

AD/Cycle
APPN
C/400
C Set ++ for OS/400
ILE
Operating System/400
Personal System/2
RPG/400
System/370
Systems Application Architecture

Application System/400
AS/400
COBOL/400
IBM
OS/400
PS/2
SAA
System/390
400

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

About Finance Communications Programming (SC41-5449)

This book contains the commands and procedures for setting up and using both the intersystem communications function (ICF) finance support and the non-ICF finance support for the AS/400 system.

Throughout this book, the terms “controller” and “processor” are used interchangeably depending on the machine type.

For a list of publications related to this book, see the “Bibliography.”

Who Should Use This Book

This book is intended for system administrators, system operators, and application programmers.

For more information on basic communications you can also refer to the *Discover/Education** course in the communications module. The *Discover/Education* course is separately orderable.

Before you use the material in this book, you should be familiar with the book, *CL Programming*. This book contains a wide-ranging discussion of AS/400 system topics, including a general discussion of objects and libraries and control language (CL) programming, as well as a discussion on controlling flow and communications. To write your transaction processing applications, you must know the programming language of the system and how to enter and create a program at an AS/400 work station.

Depending on the interface you use to communicate between the AS/400 system and the attached finance devices, you need to understand varying amounts of the information found in the *Communications Configuration* book, and the *ICF Programming* book.

Prerequisite and Related Information

For information about other AS/400 publications (except Advanced 36), see either of the following:

- The *Publications Reference* book, SC41-5003, in the AS/400 Softcopy Library.
- The *AS/400 Information Directory*, a unique, multimedia interface to a searchable database that contains descriptions of titles available from IBM or from selected other publishers. The *AS/400 Information Directory* is shipped with the OS/400 operating system at no charge.

Information Available on the World Wide Web

More AS/400 information is available on the World Wide Web. You can access this information from the AS/400 home page, which is at the following uniform resource locator (URL) address:

```
http://www.as400.ibm.com
```

Select the Information Desk, and you will be able to access a variety of AS/400 information topics from that page.

Chapter 1. Introduction to Finance Communications

This chapter is an introduction to AS/400* finance communications. It also includes an overview of the intersystem communications function (ICF) and non-ICF communications support information, and an example of a finance communications network.

Finance Communications Environment

Finance communications uses high-level language operations and communications functions that allow you to communicate between an AS/400 system and finance controllers, or between a System/370* or System/390* host system and finance controllers attached to an AS/400 system, providing a banking environment communications system. Finance communications allows programs in the supported high-level languages C Set ++ for OS/400*, (Integrated Language Environment (ILE) C/400*, ILE COBOL/400*, and ILE RPG/400* languages) on an AS/400 system to communicate with IBM* 3694, 4701, 4702, 4730, 4731, 4732, and 4736 finance processors and personal computers or Personal System/2* work stations using Financial Branch System Services (FBSS) software.

Note: The FBSS software must be version 2.2 or later when using DOS.

AS/400 finance communications includes two separate methods of communication: ICF finance communications and non-ICF finance communications. Both methods communicate using the Systems Network Architecture logical unit (SNA LU) type 0 (LU0) primary protocol.

Figure 1-1 provides an overview of both the ICF and non-ICF finance communications methods.

Figure 1-1. Overview of ICF and Non-ICF Finance Communications

ICF Finance	Non-ICF Finance
Supports the following operations and functions: • Acquire • Cancel • Cancel-invite • End-of-group • End-of-session • Function-management-header • Invite • Negative-response • Release • Read • Read-from-invited-program-devices • Timer • Write • Write-with-read	Operations limited to the following operations and functions: • Acquire • Invite • Read • Read-from-invited-devices • Release • Write • Write-with-invite Use of finance control byte allows **function-management-header**[1] for the 3694 processor
Communicates through ICF file	Communicates through display file using user-defined data streams (UDDS)
Supports requester (target) and acquired (source) sessions	Supports only acquired sessions
Supports **chaining**[2] or grouping of records	Supports only single records
Verifies security information on the INIT-SELF request by using the AS/400 system	Verifies security by using a user application
Uses ICF operations and functions to communicate between applications and devices	Uses operations or passes parameters to communicate between applications and devices

Notes:

1 A record that contains control information for the data that follows.

2 A method of storing records in which each record belongs to a list or group of records and has a linking field for tracing the chain.

Finance Communications Network Example

Figure 1-2 is one example of an AS/400 finance communications network.

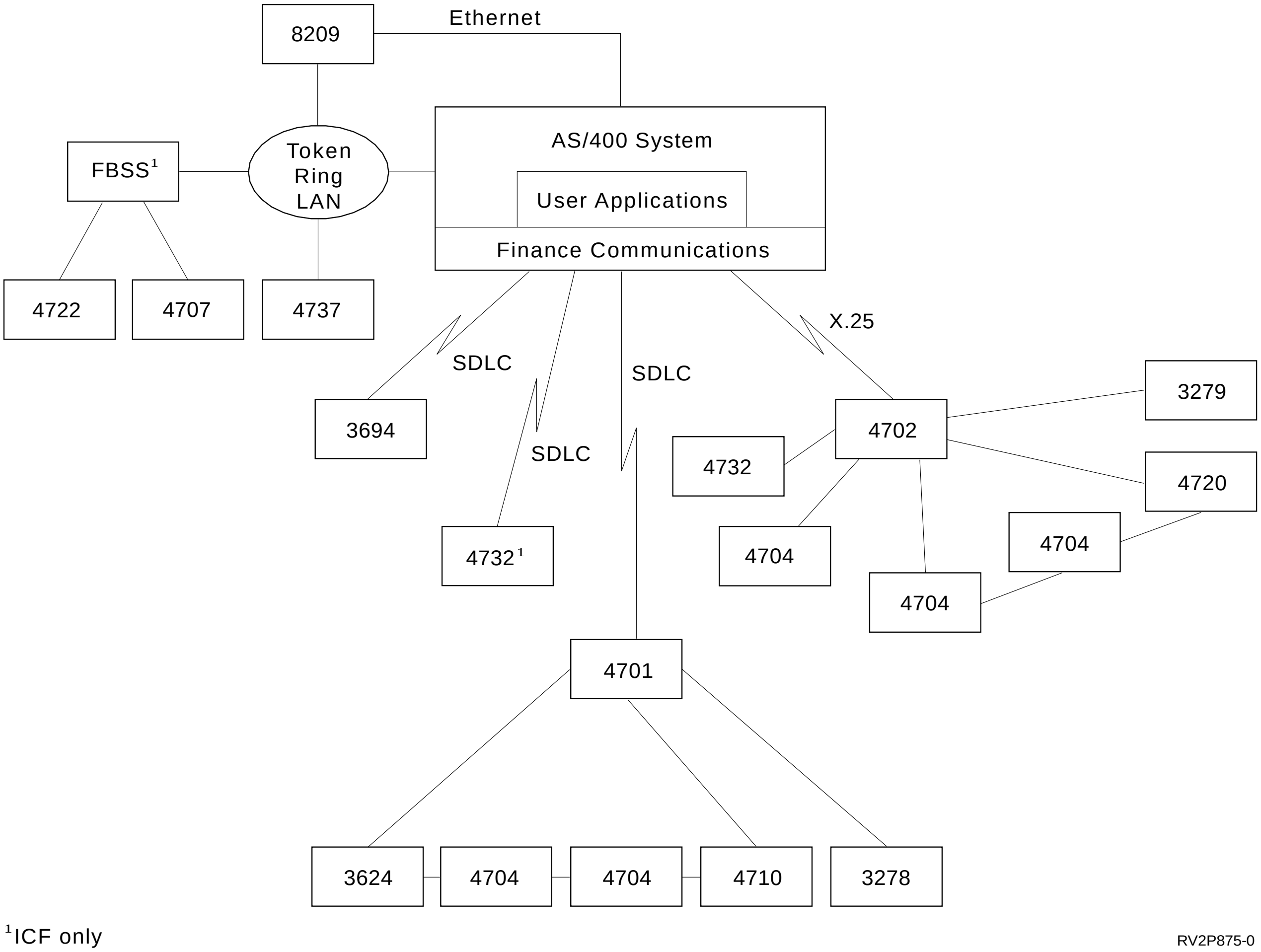


Figure 1-2. Example of an AS/400 Finance Communications Network

Finance Communications Using SNA Pass-Through Support

SNA pass-through is an AS/400 function that allows finance controllers and devices attached to the AS/400 system to communicate directly with a System/370 or System/390 host system. See "Configuring the AS/400 System for SNA Pass-Through Communications" on page 3-4 for information about configuring SNA pass-through support.

Chapter 2. Finance Communications Support

This chapter explains the functions supported by ICF and non-ICF finance communications interfaces.

Finance Functions

The types of finance controllers used with both ICF and non-ICF finance communications on the AS/400 system are:

- 3601 Finance Communications Controller
- 3694 Document Processor
- 4701 Finance Communications Controller
- 4702 Communications Processor

The types of finance controllers used only with ICF finance communications on the AS/400 system are:

- 4730 Personal Banking Machines
- 4731 Personal Banking Machines
- 4732 Personal Banking Machines
- 4736 Personal Banking Machines
- Financial Branch System Services (FBSS), including the 4737 Self-Service Transaction Station

The 3601 controller is the predecessor to the 4701 finance controller. Configure the 3601 as a 4701 controller with the appropriate exchange identifier (EXCHID) value.

4737 Self-Service Transaction Stations are configured as TYPE(*FBSS).

The 3694 processor is a programmable inscriber-reader-sorter that financial institutions use to inscribe, read, and sort magnetic-ink character recognition (MICR) documents and to capture data for subsequent host-system processing.

The 4701, 4702, and FBSS finance controllers are programmable and perform a variety of data processing transactions in a financial environment. **FBSS** is an IBM licensed program that provides extensions to the operating system of the personal computer or the PS/2* work station to support a finance industry environment. Through attached devices, tellers use finance communications for account inquiry and updating. Accountants use finance communications to maintain a record of cash flow through their businesses.

The 4730, 4731, 4732, and 4736 Personal Banking Machines are programmable and can dispense and accept money while keeping track of the transactions. Customers of financial institutions use Personal Banking Machines to transfer funds, determine account balance amounts, and make automatic payments.

The 4737 Self-Service Transaction Station is programmable and performs financial tasks such as transferring funds, determining account balance amounts, and making automatic payments.

Using finance communications, you can do the following:

- Send and receive data between the AS/400 system and the finance controller programs and describe how to present that data.
- Use SNA pass-through to provide communications between finance controllers and devices attached to the AS/400 system and a System/370 or System/390 host system.
- Communicate with 4701 and 4702 controllers using an **X.21** circuit-switching network. X.21 is the Telecommunication Standardization Sector (CCITT), formerly known as ITU-T, specification that defines connection of data terminal equipment to an X.21 network.
- Communicate with 4701, 4702, and FBSS controllers through an **X.25** packet-switching data network. X.25 is the ITU-T specification that defines the interface to an X.25 network. The AS/400 system attaches to an X.25 network using a nonswitched line through either an X.21 or X.21 bis (V.24 or V.35) physical shared boundary.
- Communicate with FBSS controllers on a token-ring network using Systems Network Architecture (SNA).
- Communicate between an FBSS controller on a token-ring network and an AS/400 system on an Ethernet network using the 8209 LAN bridge to connect the two LANs.

- Allow 3270 sessions and finance sessions to share the same X.25 virtual circuit.
- Allow attached 4701, 4702, and FBSS finance controllers to share the X.25 line with any AS/400 session types possible for X.25 communications.
- Attach any finance controller to your AS/400 system using the **synchronous data link control (SDLC)** protocol. SDLC is a type of communications line control that uses commands to control the transfer of data over a communications line. SDLC can have one of the following configurations:
  - Nonswitched point-to-point
  - Nonswitched multipoint
  - Switched point-to-point, using the following connection methods:
    - Automatic dial (uses a serial automatic dial modem).
    - Automatic call (two-line EIA 366/V.25) support. You need RPQ 843567 for the 9406 System Unit and RPQ 843568 for the 9404 System Unit.
    - Automatic answer.
    - Manual dial.
    - Manual answer.

  The finance controllers can share the same SDLC multipoint communications line with any other session type using an SDLC primary role on the AS/400 system.

  Note: The AS/400 system allows a maximum of 254 controllers to be attached to a multipoint line.

Concurrent Sessions

The AS/400 system supports concurrent operation of 3270 displays, printers, and finance work stations on one controller. However, the controller must also support concurrent LU0 and 3270 (LU1, LU2, and LU3) device operation. For 4701 and 4702 controllers, the type and the application of each device attached to the controller must be defined in the controller program generator (CPGEN) file on the controller. The CPGEN file determines what session type the device will be using.

The 3270 device support includes the IBM 3277, 3278, and 3279 displays and the IBM 3287 printer. For more information about the 3270 work station support, see the topic on using the 3270 remote attachment support in the *Remote Work Station Support, SC41-5402*.

Security

The security provided on the AS/400 system controls the use of communications device descriptions and commands used with the device descriptions. Security also controls access to programs and objects used by programs.

For finance-specific considerations, see Chapter 7, “Finance Considerations,” and for general system security information, see the *Security – Reference* book.

Error Handling and Recovery

Application programs use major and minor error return codes to handle error conditions. Applications written in high-level languages can access return codes to help diagnose problems. The C Set ++ for OS/400, ILE COBOL/400, and ILE RPG/400 languages provide language-defined file values for status that can be used together with the major and minor codes. Messages are entered in the job log to identify the error that occurred.

Note: The ILE C/400 language does not support file status values. However, you can look at the value of the global variable in the errno.h header file to check for any I/O exceptions that may occur and to retrieve any exception data associated with the error.

If an error results in a notify message, it is recoverable with little or no operator involvement.

For error recovery specific to ICF finance, see Appendix B, “Return Codes, Messages, and Sense Codes.” For error recovery specific to non-ICF finance, see Chapter 6, “Writing Non-Intersystem Communications Function Application Programs.”

For information describing recovery from line errors, see the *Communications Management* book.

System Monitor Session

The system monitor is a controller program used to perform service, configuration, and debugging functions on 4701 and 4702 finance controllers. Local location address (LOCADR) 01 is reserved for this program. When using finance communications, your ICF application program communicates with the system monitor program by using a device configuration specified as LOCADR (01) and a finance type of *FNCICF.

4701 Finance Controller Diskette Download: The AS/400 system supports the downloading of an operating diskette image for an 8-inch diskette. The operating diskette image for the 4701 finance controller must be created on a System/370 computer and loaded on the AS/400 system. After the diskette image is loaded on the AS/400 system, use the diskette download support to send the diskette image to the controller through the system monitor session. For more information about this feature, see Appendix D, "4701 Finance Controller Diskette Download."

Finance Interfaces

Operating System/400* (OS/400*) finance communications provides support allowing application programs to communicate with finance controllers. The application programmer can use ICF finance or non-ICF finance to access this support.

Regardless of the interface you choose, you must supply the following:

- Finance controller application programs
- AS/400 application programs to process financial transactions

You can write application programs on the AS/400 system using the supported high-level language licensed programs.

Intersystem Communications Function Finance Support

Specify TYPE(*FNCICF) in the device description to identify that you are using ICF finance. Application programs use an ICF file to send and receive data. The ICF file contains the file description identifying the record formats used by the application programs. For more information about writing ICF finance applications, see Chapter 5, "Writing Intersystem Communications Function Finance Application Programs."

Non-Intersystem Communications Function Finance Support

The non-ICF finance communications support is separate from and completely independent of ICF finance communications support. Specify 4704, 3624, or 3694 for the TYPE parameter in the device description to identify that you are using non-ICF finance. Non-ICF finance uses a display file to send and receive data. The display file contains the file description identifying the record formats used by the application program and finance support.

You can choose one of the following for non-ICF finance communications:

- The Submit Finance Job (SBMFNCJOB) command with the Work with Table (WRKDEVTBL, WRKUSRTBL, or WRKPGMTBL) commands.

  For information about the Submit Finance Job and Work with Table commands, see "Using the Submit Finance Job Command" on page 6-1.

- Finance I/O Manager (FIOM) alone or with either the SBMFNCJOB command or the user-defined data stream (UDDS). FIOM is the recommended alternative to UDDS. For more information about FIOM, see "Using Finance Input/Output Manager" on page 6-7.

- UDDS. For more information about UDDS, see "Using User-Defined Data Stream" on page 6-14.

Chapter 3. Configuring Finance Support

This chapter describes the commands used to configure your AS/400 system for finance communications. See the *Communications Configuration* book for a complete description of AS/400 configuration commands and parameters.

Configuring the AS/400 System for Finance Communications

The following commands allow you to create, change, display, print, and delete the line, controller, and device descriptions.

Configuring the Line Description

An AS/400 line description contains configuration information, such as the port number for communications lines, network characteristics, and physical information needed by the system. Create a line description for each communications line used to communicate with a finance controller.

Use the following commands to create, change, display, print, or delete line descriptions:

CRTLINSDLC
The Create Line Description (SDLC) command allows you to create a line description for each SDLC communications line used to communicate with the finance controllers. Each line should be created as the SDLC primary role.

CRTLINX25
The Create Line Description (X.25) command allows you to create a line description for each X.25 communications line used to communicate with the 4701, 4702, and Financial Branch System Services (FBSS) controllers.

CRTLINTRN
The Create Line Description (Token-ring) command allows you to create a line description for a token-ring network used with an FBSS controller.

CRTLINETH
The Create Line (Ethernet) command allows you to create a line description for an Ethernet line used to communicate with finance controllers over an 8209 LAN Bridge that connects the Ethernet line to a token-ring LAN to which the controller is attached.

CHGLINSDLC
The Change Line Description (SDLC) command allows you to change the configuration parameters for an SDLC communications line used to communicate with finance controllers.

CHGLINX25
The Change Line Description (X.25) command allows you to change the configuration parameters for an X.25 communications line used to communicate with finance controllers.

CHGLINTRN
The Change Line Description (Token-ring) command allows you to change the configuration parameters for a token-ring network used to communicate with finance controllers.

CHGLINETH
The Change Line Description (Ethernet) command allows you to change the configuration parameters for an Ethernet line used to communicate with finance controllers over an 8209 LAN Bridge that connects the Ethernet line to a token-ring LAN to which the controller is attached.

DSPLIND
The Display Line Description command allows you to display or print a line description.

DLTLIND
The Delete Line Description command allows you to delete a line description.

Configuring the Controller Description

A finance controller description defines configuration parameters unique to each finance controller, such as the station address to the system.

These controller descriptions are linked to the configured nonswitched line description when you create the descriptions. Controller descriptions for switched connections are linked to the first available line in the configured switched line list when the call is made.

Use the following commands to create, change, display, or delete finance controller descriptions:

CRTCTLFNC
The Create Controller Description (Finance) command allows you to create a finance controller description for each finance controller with which your AS/400 system is communicating.

CHGCTLFNC
The Change Controller Description (Finance) command allows you to change the configuration parameters for a finance controller description.

DSPCTLD
The Display Controller Description (Finance) command allows you to display or print a controller description.

DLTCTLD
The Delete Controller Description (Finance) command allows you to delete a controller description.

Configuring the Device Description

Device descriptions contain the logical unit (LU) address and device type of each device on a finance controller on the AS/400 system. (The LU address is known as the local location address.)

The local location address must match the address you used to identify a particular physical device. The local location address (LOCADR) parameter you specify can be a value from hex 02 through 3C for the non-ICF finance communications method and hex 01 through FF for ICF finance.

The device description also defines which finance communications method, ICF or non-ICF, is used. If you use ICF with a finance controller, you can also define a system monitor session by specifying a local location address of 01 and a finance type of *FNCICF.

You can create device descriptions for the following finance types:

- ICF finance device description (*FNCICF)
- 3624 Consumer Transaction Facilities (CTF)
- 3694 Document Processors
- 4704 Finance Communication Display Stations

Note: When you define a device as *FNCICF, the device can be any supported finance device.

Use the following commands to create, change, display, print, or delete device descriptions:

CRTDEVFNC
The Create Device Description (Finance) command allows you to create a device description for a finance device that will be linked to a finance controller. If you use ICF finance communications, you must specify *FNCICF as the device type (TYPE) parameter in the CRTDEVFNC command. If you use non-ICF finance, specify the specific device number, such as 4704, 3624, and 3694, in the TYPE parameter.

CRTDEVDSP
The Create Device Description (Display) command allows you to create a device description for each 3270 family display linked to each finance controller. Specify the class as a remote (*RMT) device. You can attach 3277, 3278, 3279, and 3287 device types to a finance controller.

CRTDEVPRT
The Create Device Description (Printer) command allows you to create a device description for each 3287 printer attached to the finance controller. Specify the class as a remote (*RMT) device.

CHGDEVFNC
The Change Device Description (Finance) command allows you to change the configuration parameters for each finance controller.

CHGDEVDSP
The Change Device Description (Display) command allows you to change the configuration parameters for each display attached to the finance controller.

CHGDEVPRT
The Change Device Description (Printer) command allows you to change the configuration parameters for each printer attached to the finance device.

DSPDEVD
The Display Device Description command allows you to display or print a device description.

DLTDEVD
The Delete Device Description command allows you to delete a device description.

You can define up to the following maximum number of devices for each controller:

Figure 3-1. Number of Devices per Controller

Controller	Maximum Devices	Maximum Active at Once
4701	255	120
4702	255	120
FBSS	255	240
4731, 4732, 4736	2	2
3694	4	4
4730	3	3

Only devices specified as TYPE(3694) or TYPE(*FNCICF) can be attached to the 3694 controller. Only devices specified as TYPE(*FNCICF) can be attached to the 4730, 4731, 4732, or 4736 controllers. An FBSS controller can have *FNCICF, 3277, 3278, 3279, and 3287 devices attached. You can configure any of the following device types for a 4701 or 4702 controller: 3277, 3278, 3279, 3287, 3624, 4704, and *FNCICF.

Note: If you are using non-ICF finance communications, you can attach a 4710 or 4720 printer to the AS/400 system by configuring the printer as a 4704 device. If you are using ICF finance communications, configure the printer as a *FNCICF type.

The device type you specify in the device description determines the format of the data stream sent to the finance controller. The data stream can be one of the following:

- LU0 for device type 3624, 3694, 4704, or *FNCICF
- LU1 for device type 3287
- LU2 for device type 3277, 3278, or 3279

Note: If you specify a 32XX device, the 4701, 4702, and FBSS controllers must be capable of supporting 3270 logical unit 1 (LU1) and 3270 logical unit 2 (LU2) data streams.

Using 3270 Devices

If your 4701, 4702, or FBSS finance controller can support the 3270 family of displays and printers, you can also attach those devices to your finance controller.

The AS/400 system requires that any device configured as a 3278 or a 3279 must accept extended data streams. Be sure that the application program running on the 4701, 4702, or FBSS controller has extended data stream support. If it does not have extended data stream support, configure each device as a type 3277. LU0 data streams allow printer data to be included with display data. LU1 data streams are sent to printers. LU2 data streams require you to create 3270-type application displays. Be aware that translation and emulation can make the LU2 approach run slower than the LU0 support.

You can program a 4701, 4702, or FBSS controller to do 3270 emulation. You can also configure a finance device as a 3270 display, or you can attach 3278, 3279, and 3287 devices to the controller. You must use an LU2 emulation package on the controller regardless of which devices you use. Using this mode of operation, you can configure the controller as follows:

- As an FBSS controller with 3270 emulation or 3287 printer emulation plus FBSS-supported finance devices attached
- As a 4701 or 4702 controller with 3277, 3278, 3279, 3287 devices and, optionally, 3624 and 4704 devices attached
- As a 3274 controller with only 3270 devices attached

For more information about 3270 remote attachment support, see the *Remote Work Station Support SC41-5402* book.

You must vary on the line, controller, and device descriptions for finance support before any AS/400 application program can communicate with a finance controller. For information about the vary on process, see Chapter 4, “Running Finance Support.” For more information about the communications configuration process and the commands used for configuration, see the *Communications Configuration* book.

Configuring the AS/400 System for SNA Pass-Through Communications

SNA pass-through support allows finance controllers and devices attached to an AS/400 system to communicate directly with a System/370 or System/390 host system using the SNA LU0, LU1, or LU2 protocols.

You configure the AS/400 system for SNA pass-through communications by creating line, controller, and device descriptions. These objects describe the connections between the AS/400 system and the host system and between the AS/400 system and the finance controller and devices.

Line Descriptions

Two line descriptions must be created, one describing the connection between the finance controller and the AS/400 system, the other describing the connection between the AS/400 system and the host system. The line descriptions need not be of the same type; for example, the finance controller can be attached to the AS/400 system using a token-ring line while the AS/400 system is connected to the host system using an SDLC line.

The following commands may be used to create the line descriptions:

- CRTLINIDLC
- CRTLINETH
- CRTLINSDLC
- CRTLINTRN
- CRTLINX25

Notes:

1. ISDN communications (using IDLC or X.25 lines) requires additional configuration of a network interface description and connection list. See the *ISDN Support* book for more information and examples of ISDN communications.
2. An FBSS controller attached to a token-ring network can be connected to an AS/400 system on an Ethernet network by using the 8209 LAN Bridge. The *LAN and Frame Relay Support* book contains configuration examples and considerations for bridged environments.

Controller Descriptions

Two controller descriptions must be created for SNA pass-through communications. A **host controller**, describing the host system, must be created using the CRTCTLHOST command. An **APPC controller** must also be created to represent the finance controller that will communicate with the host system. Use the CRTCTLAPPC command to create the APPC controller description.

For finance controller types 3694, 4701, 4702, 4730, 4731, 4732 or 4736, set the TYPE parameter on the CRTCTLAPPC command to the controller type. Specify *NO for the APPN parameter to indicate that the controller does not use APPN*—LU 6.2. For instance, for a 3694 controller, specify: TYPE(3694) APPN(*NO).

For a PS/2* work station running FBSS with the DOS operating system, specify: TYPE(*FBSS) for APPC controllers.

For a PS/2* work station running FBSS with the OS/2* operating system, specify either APPN(*YES) or TYPE(*BLANK). If you specify TYPE(*BLANK), you must set the SSCPID parameter to a value other than zero.

Device Descriptions

At least two device descriptions must be created; a finance device and an SNA pass-through device. The SNA pass-through device description associated with the host logical unit is created using the CRTDEVSNPT command and must specify SNPTCLS(*UP). The finance device description is created using the CRTDEVFNC command and must specify DEVCLS(*SNPT) and TYPE(*FNCICF). 3270 device descriptions attached to finance controllers can also be configured as SNA pass-through devices using the CRTDEVDSP and CRTDEVPRT commands.

Note: Finance controllers, attached finance devices, and attached 3270 devices that are configured to use SNA pass-through support must be reconfigured to allow communications between the finance controller and the AS/400 system. To reconfigure for communications between the finance controller and the AS/400 system, you must vary off the configuration objects associated with the finance controller (line description, APPC controller, and devices) and reconfigure the controller using the CRTCTLFNC command. If the connection uses a switched SDLC line or X.25 communications, you must also delete the APPC controller description.

Associating Finance and SNA Pass-Through Devices

SNA pass-through configuration requires that each finance device be paired with an SNA pass-through device. The logical association of the devices can be done in either of two ways:

- Specify the SNA pass-through device name: The SNA pass-through device (SNPTDEV parameter) on each device description can specify the name of the other device description. That is, the finance device description SNPTDEV parameter specifies the name of the SNA pass-through device; the SNA pass-through device SNPTDEV parameter specifies the name of the finance device.
- Specify a group of SNA pass-through devices: The SNA pass-through group name (SNPTGRP parameter) on each device description can specify the name of a group of devices. For example, the finance device description can specify the name of a group of SNA pass-through devices. SNA pass-through groups are defined in configuration list entries for the QSNAPASTHR configuration list.

  This method allows the system to associate the finance device with any SNA pass-through device defined in the group. When the finance device is varied on, the system will select the first available SNA pass-through device listed in the group specified on the SNPTGRP parameter.

  Only one configuration list of type *SNAPASTHR is allowed on the system at one time. Use the Add Configuration List Entries (ADDCFGLE) command to define SNA pass-through groups within the QSNAPASTHR configuration list.

A finance device that specifies DEVCLS(*SNPT), SNPTDEV(*NONE), and SNPTGRP(*NONE) can be associated with an SNA pass-through device only if the SNA pass-through device either specifies the name of the finance device on the SNPTDEV parameter or specifies an SNA pass-through group (SNPTGRP parameter) that includes the finance device.

See the book, *Remote Work Station Support, SC41-5402* for more information and examples of configuring SNA pass-through support.

Using the Work with Table Commands to Configure Non-Intersystem Communications Function (Non-ICF) Finance

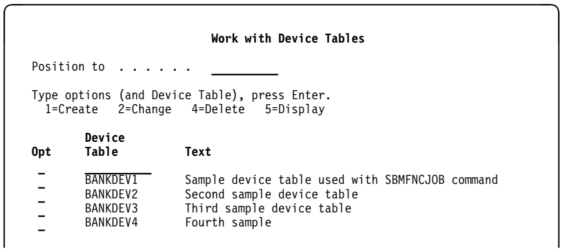
You can use the following commands to configure the Submit Finance Job (SBMFNCJOB) command for non-ICF finance communications. For more information about the syntax or parameters for these commands, see the *CL Reference* book.

Work with Device Table Command

A finance device table is a list of finance devices that can be acquired by the finance job using the SBMFNCJOB command. Use the Work with Device Table (WRKDEVTBL) command to create, change, delete, or display finance device tables.

Device table updates can be used by any finance job that is submitted after all changes are done.

When you enter the WRKDEVTBL command, the Work with Device Tables display appears:



```
                         Work with Device Tables

 Position to  . . . . . .   __________

 Type options (and Device Table), press Enter.
   1=Create   2=Change   4=Delete   5=Display

         Device
 Opt     Table        Text

  _      __________
         BANKDEV1     Sample device table used with SBMFNCJOB command
  _      BANKDEV2     Second sample device table
  _      BANKDEV3     Third sample device table
  _      BANKDEV4     Fourth sample
  _
```

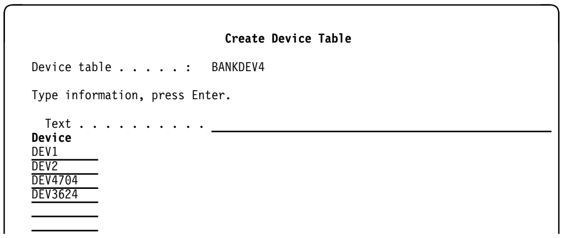
Create a new table
Create one new table at a time by typing a 1 in the first blank line under the *Opt* column and typing in the new name.

Change a table
Select a table to change by typing a 2 next to the table name in the list, or by typing a 2 in the first blank line under the *Opt* column and typing in the name.

Delete one or more tables
Type a 4 next to each table you want to delete.

Display a table
Type a 5 next to the name of the table you want to display.

The Create Device Table display appears when you select option 1 (Create) on the Work with Device Tables display:



```
                        Create Device Table

Device table . . . . . :   BANKDEV4

Type information, press Enter.

  Text . . . . . . . . . . ________________________________________
Device
DEV1_____
DEV2_____
DEV4704__
DEV3624__
_________
_________
```

To add a device name, enter the device on one of the input lines supplied. The device name you add should have a device description (created using the CRTDEVFNC command) with the same name (DEVD parameter) and a device type of 3624 or 4704. Messages appear warning you that the device description does not yet exist, or that the device type is not 3624 or 4704, but the device name is still added to the table. You can also specify up to 50 characters of descriptive text for the new table on the line supplied.

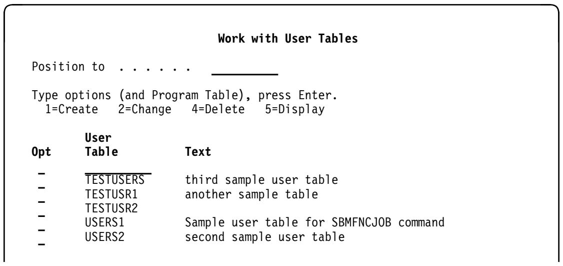
Work with User Table Command

The finance user table is a list of approved finance user IDs. The finance job uses these IDs to approve user IDs sent in the data stream with the INIT-SELF request from the finance controller. Finance user IDs are not related to user profiles.

Use the Work with User Table (WRKUSRTBL) command to create, change, delete, or display any number of user tables that define the finance user IDs. Each table must have a unique name.

User table updates can be used by any finance job that is submitted after all changes are done.

When you enter the WRKUSRTBL command, the Work with User Tables display appears:



```
                        Work with User Tables

Position to  . . . . . .   _________

Type options (and Program Table), press Enter.
  1=Create   2=Change   4=Delete   5=Display

       User
Opt    Table         Text

 _     TESTUSERS     third sample user table
 _     TESTUSR1      another sample table
 _     TESTUSR2
 _     USERS1        Sample user table for SBMFNCJOB command
 _     USERS2        second sample user table
 _
```

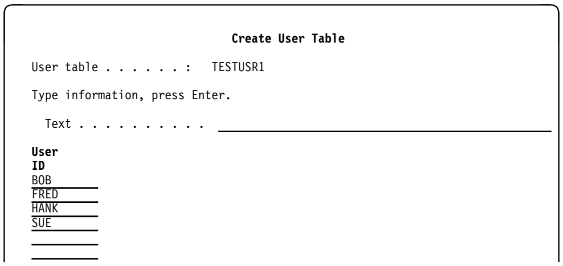
Create a new table
Create one new table at a time by typing a 1 in the first blank line under the *Opt* column and typing in the new name.

Change a table
Select a table to change by typing a 2 next to the table name in the list, or by typing a 2 in the first blank line under the *Opt* column and typing in the name.

Delete one or more tables
Type a 4 next to each table you want to delete.

Display a table
Type a 5 next to the name of the table you want to display.

The Create User Table display appears when you select option 1 (Create) on the Work with User Tables display:



```
                        Create User Table

User table . . . . . . :   TESTUSR1

Type information, press Enter.

  Text . . . . . . . . . .  ________________________________________

User
ID
BOB______
FRED_____
HANK_____
SUE______
_________
_________
```

To add a user table, enter the user ID on one of the input lines supplied. You can specify up to 50 characters of descriptive text for the new table on the line supplied.

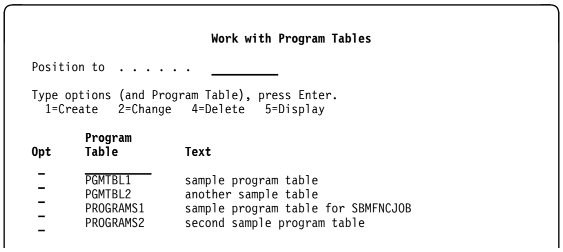
Work with Program Table Command

The finance program table is a list of your AS/400 transaction-processing application programs, with their associated program IDs. These program IDs are included with the data in the transaction request by a finance controller. The program ID is then used to call the correct AS/400 application program to process the transaction.

Use the Work with Program Table (WRKPGMTBL) command to create, change, delete, or display, any number of program tables. Each table must have a unique name.

Program table updates can be used by any finance job that is submitted after all changes are done.

When you enter the WRKPGMTBL command, the Work with Program Tables display appears:



```
                        Work with Program Tables

 Position to  . . . . . .   __________

 Type options (and Program Table), press Enter.
   1=Create   2=Change   4=Delete   5=Display

        Program
 Opt    Table         Text

  _     __________
  _     PGMTBL1       sample program table
  _     PGMTBL2       another sample table
  _     PROGRAMS1     sample program table for SBMFNCJOB
  _     PROGRAMS2     second sample program table
```

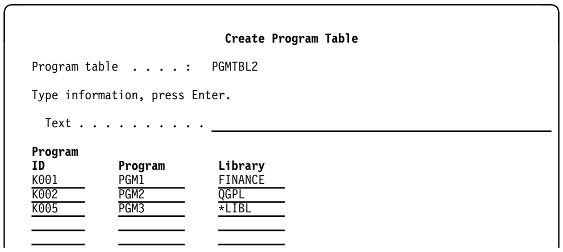
Create a new table
Create one new table at a time by typing a `1` in the first blank line under the *Opt* column and typing in the new name.

Change a table
Select a table to change by typing a `2` next to the table name in the list, or by typing a `2` in the first blank line under the *Opt* column and typing in the name.

Delete one or more tables
Type a `4` next to each table you want to delete.

Display a table
Type a `5` next to the name of the table you want to display.

The Create Program Table display appears when you select option 1 (Create) on the Work with Program Tables display:



```
                          Create Program Table

 Program table  . . . . :   PGMTBL2

 Type information, press Enter.

   Text . . . . . . . . . .  ____________________________________________

 Program
 ID          Program       Library
 K001______  PGM1________  FINANCE_____
 K002______  PGM2________  QGPL________
 K005______  PGM3________  *LIBL_______
 __________  ____________  ____________
 __________  ____________  ____________
```

Enter new program IDs and program names by typing the following information on one of the extra input lines supplied:

```
PROGRAM ID (required)
PROGRAM NAME (required)
LIBRARY NAME (optional - *LIBL is default)
```

A message appears warning that the program or the library does not exist. However, entries are still added to the table. You can also specify up to 50 characters of descriptive text for the new table on the line supplied.

Chapter 4. Running Finance Support

This chapter contains information you need to run your network, including information about the Vary Configuration (VRYCFG) command and the Work with Configuration Status (WRKCFGSTS) command.

Varying Finance On and Off

You must define and vary on the communications configurations before using your communications applications. You can specify the configurations to be automatically varied on at initial program load (IPL) when you create your configurations. You can also use the Vary Configuration (VRYCFG) command or the Work with Configuration Status (WRKCFGSTS) command to vary on and off the appropriate network interface line, controller, and device descriptions.

Using the Vary On Configuration Command

When using the VRYCFG command, specify the following parameters:

CFGOBJ
Specifies the name of the line, controller, or device description to be varied on or off, or a list of names of configuration objects of the *same* description type (for example, a list of line description names).

CFGTYPE
Specifies the type of configuration to be varied on or off. This is a required parameter. Valid entries are:

- *CTL: Controller configuration
- *DEV: Device configuration
- *LIN: Line configuration

STATUS
Specifies the status of the configuration object. Valid entries are:

- *ON: Object is varied on.
- *OFF: Object is varied off.

  Note: All devices must be varied off before the attached controller can be varied off. All controllers must be varied off before the associated line can be varied off. (This can be done by using the RANGE parameter.) A device can be varied off only when it is not allocated for an active job.

RANGE
Specifies what group of configuration objects should be varied on or off. Valid entries are:

- *NET: All downline configuration objects are varied on or off. Downline objects are devices attached to a controller and controllers that are attached to a communications line in a communications configuration.
- *OBJ: Only the specified object is varied on or off.

  Note: When *NET is specified, the system does the vary off procedures in the appropriate order. The objects include the configuration object or objects specified and their attached configuration objects only. For example, for a communications line, the attached objects are controllers and devices; for a controller, the attached objects are devices. Devices do not have attached objects.

VRYWAIT
Specifies whether the Ethernet, token-ring, X.25, or switched SDLC line description is varied on asynchronously or synchronously. The VRYWAIT parameter specifies how long the system must wait for synchronous vary on to be completed. Once completed, the ICF file can be opened and the session acquired.

- *CFGOBJ: Specifies that the VRYWAIT parameter value on the line description is to be used.
- *NOWAIT: Specifies that the line is to be varied on asynchronously. This signals the system not to wait for vary on completion.

- *vary-on-wait:* Specifies a value ranging from 15 through 180 seconds in 1-second intervals. The system waits until either the line is varied on or the timer expires before completing the VRYCFG command.

ASCVRYOFF

Specifies whether the vary off is asynchronous. This parameter is not allowed when the STATUS(*ON) parameter is specified. Valid entries are:

- <u>*NO</u>: The vary off is synchronous.
- *YES: The vary off is asynchronous.

Using the Work with Configuration Status Command

Using this command, you access the Work with Configuration Status display.

On the Work with Configuration Status display, you can choose the following options:

- Vary on (option 1): Varies on the line, controller, or device and all of the attached controllers or devices. This is the same as using the Vary Configuration (VRYCFG) command with STATUS(*ON).
- Vary off (option 2): Varies off the line, controller, or device and all of the attached controllers or devices. This is the same as using the Vary Configuration (VRYCFG) command with STATUS(*OFF). You may vary off devices only if they are not allocated to an active job. Jobs can be canceled if you need to vary off a device.

For the syntax of the VRYCFG and WRKCFGSTS commands, see the *CL Reference* book. For more information about the VRYCFG and WRKCFGSTS commands, see the *Communications Management* book.

Defining Communications Entries

IBM supplies two subsystem configurations to support ICF finance communications. These are QBASE, the **controlling subsystem**, and QCMN, the communications subsystem. The controlling subsystem is the interactive subsystem through which the system operator controls the system. The communications subsystem is used when QCTL is the controlling subsystem. If either of these subsystems is active, program start requests may be accepted for finance communications.

The AS/400 system considers communications controllers to be another source of work for the subsystem. Therefore, you must use a communications entry in the subsystem description to identify the devices from which work (the program start request) can be received by the subsystem. To create subsystem descriptions using the Create Subsystem Description (CRTSBSD) command, see the *CL Reference* book.

Default communications entries are shipped with the subsystem. QBASE and QCMN subsystem descriptions are shipped with a default communications entry specified as DEV(*ALL) and MODE(*ANY) to allow program start requests from any ICF communications type. With the following commands, you can change these entries:

- Add Communications Entry (ADDCMNE): Adds a device or devices to the subsystem.
- Remove Communications Entry (RMVCMNE): Removes a device or devices from a subsystem.
- Change Communications Entry (CHGCMNE): Changes an existing entry for a device.

Note: These commands cannot change the communications entries of the subsystem if the subsystem is active.

For more information about these commands, see the *Communications Management* book. For more information about managing your subsystems, see the *Work Management* book.

Starting the Finance Subsystem

The appropriate subsystem must be started if you want to use finance devices.

Non-Intersystem Communications Function:

For non-ICF finance, the finance subsystem (QFNC) must be started by using the Start Subsystem (STRSBS) command before you can use the Submit Finance Job (SBMFNCJOB) command.

Intersystem Communications Function Finance: For ICF finance, the appropriate subsystem must be started before program start requests can be received from the finance controller. Use the Start Subsystem (STRSBS) command to start a subsystem to be used for ICF finance communications.

Chapter 5. Writing Intersystem Communications Function Finance Application Programs

This chapter describes how an application program uses the AS/400 system, intersystem communications function (ICF) support, and finance communications to communicate with a finance controller. You can code the program using supported, high-level language programs to do the following functions:

- Start a session by opening a file and acquiring a program device, either explicitly or implicitly.
- Send and receive information by writing or reading to a program device.
- End a session by releasing the program device and closing the file.

Notes:

1. Check the major and minor return codes, as well as any response indicators, after each operation that your program issues.

This chapter also includes a description of the read and write operations that specify a record format containing specific communications functions. Record formats can be defined using the data description specifications (DDS) keywords or system-supplied formats. For a list of DDS keywords and system-supplied formats, see Appendix A, “Language Operations, DDS Keywords, and System-Supplied Formats,” or the *DDS Reference* book.

To help you write and use programs on the AS/400 system, see Appendix E, “Intersystem Communications Function Finance Example Programs.”

Using Intersystem Communications Function File Commands

An Operating System/400 intersystem communications function (ICF) file must be created before your application can use finance communications. For more information about the ICF file, see the book, *ICF Programming*.

The ICF file is a system object type of *FILE with a specific user format. This format consists of a set of commands and operations. The commands allow you to manage the file attributes and allow you to create, delete, change, and display the file description. The operations allow a program to use the file.

The following commands are valid for finance communications:

CRTICFF
The Create ICF File command allows you to create an ICF file specifying file level attributes.

CHGICFF
The Change ICF File command allows you to change the file attributes of the ICF file.

OVRICFF
The Override ICF File command allows you to temporarily change the file attributes of the ICF file at run time. These changes are in effect only for the duration of the job and do not affect other users of the file.

DLTOVR
The Delete Override command deletes the effect of the OVRICFF command.

DLTF
The Delete File command deletes the file from the system and frees the storage space allocated to that file.

DSPFD
The Display File Description command displays information about the attributes of the device file.

DSPFFD
The Display File Field Description command displays field-level information for a device file.

ADDICFDEVE
The Add ICF Device Entry command adds a program device entry, with the specified device name and attributes, to the file. You can use this command many times to

add multiple program devices to the same file.

CHGICFDEVE
The Change ICF Device Entry command changes the program device entry defined with the ADDICFDEVE command.

OVRICFDEVE
The Override ICF Device Entry command overrides the attribute specified in the ADDICFDEVE command, or temporarily associates the specified program device name and attributes with the file.

This command differs from the ADDICFDEVE command in that it does not permanently change the ICF file. The association between the program device entry and the file exists only for the job in which the command runs. You can use this command many times to override multiple program device entries in the file.

DLTOVRDEVE
The Delete Override Device Entry command deletes the effect of the OVRICFDEVE command.

RMVICFDEVE
The Remove ICF Device Entry command removes one or more program device entries from the file.

DSPOVR
The Display Override command displays overrides created by the OVRICFF file command.

The program device entry associates a program device name with a device description.

The ADDICFDEVE, CHGICFDEVE, and OVRICFDEVE commands use the following parameters for finance communications:

FMTSLT
Determines the format selection option. This parameter indicates the type of processing used to determine what record format to use on input operations. Finance communications supports only the program (*PGM) and record identification (*RECID) values of the FMTSLT parameter. For more information about the FMTSLT parameter, see the book, *ICF Programming*.

PGMDEV
Specifies the program device name being defined (the name used by the program to do the operations). The program device name must be unique throughout all entries in the file. You can map two or more different program device names to the same communications configurations.

PGMDEV is a required parameter.

RMTLOCNAME
Specifies the **remote location** name associated with the program device. The remote location name specifies another system with which your system can communicate in an SNA network. A remote location is associated with any device description that contains the same remote location name.

You cannot configure the same remote location name in more than one device description.

The system uses the remote location name to select the device description. The RMTLOCNAME parameter does not need to exist at the time you define the program device entry; however, it must exist as a part of the device description on the system when the program is acquired. If your program is started by a program start request, a RMTLOCNAME of *REQUESTER must be used.

RMTLOCNAME is a required parameter.

Starting a Session

Before your program can communicate with the controller program, you must establish an ICF communications session. A communications session is a logical connection between two systems through which a finance controller program communicates with a program on an AS/400 system. This communications session is established with an open/acquire operation and is ended with a release operation or an end-of-session function or a close operation.

Starting a Session with a 47xx or Financial Branch System Services Controller

You can start the session with the 47xx or FBSS controllers in one of two ways: by the controller sending an INIT-SELF command and then sending a program start request, or by your AS/400 program issuing an ICF Acquire operation.

The following explains how to start a controller-initiated session:

1. Start the SNA session from the controller by sending an INIT-SELF command (considered a logon request). The AS/400 system sends a BIND command to start the SNA session. On a secure AS/400 system, the BIND command is sent only if the user data field of the received INIT-SELF command contains a valid user ID and password.

   The user data field may also contain a default library name. If the *library name* field is included, that library is used to search for a program that is specified in a program start request sent later by the controller (if a specific library is not specified on that request).

2. Start the ICF session and transaction by sending a program start request using an *EXEC statement. The AS/400 finance support starts the program specified on the *EXEC statement. The target program issues the ICF Acquire operation and establishes the session and transaction.

The following explains how to start an AS/400-initiated session:

1. Start the SNA session from your AS/400 source program by issuing an ICF Acquire operation. This is known as an unsolicited start session request because the controller does not request the session.
2. The session and transaction are established automatically.

Starting Your Program: Your finance programs can be started by the controller sending a program start request or by the use of a manually started program.

Program Start Request from the Controller: Typically, your program is started after a program start request is received from the controller. The program start request contains an *EXEC request statement with a program name, an optional library name, and optional program parameters.

Note: Program start requests can be received after the AS/400 system sends a BIND command and before the AS/400 system sends an UNBIND command.

Additional program start requests received while a transaction program is running are treated as user data.

The syntax of the program start request is:

```
*EXEC program name[,library name] [optional data]
```

Program name
Name (from 1 to 10 characters in length) of the program to be started. The program name must start in position 7. A blank must be in position 6.

Library name
Name (from 1 to 10 characters in length) of the library containing the program to be started. This parameter is optional. If this name is specified on both the INIT-SELF command and the *EXEC request statement, this library name is used for the program start request. If the library name is not specified in either place, the program must exist in a library of the job description specified in the communications entry of the subsystem in which it runs.

Optional data
From 1 to 512 bytes of optional data following the program name or an optional library name. A blank must be included between the program name, or the optional library name, and the data.

AS/400 System Started Program: You can start the program with an AS/400 command or a command from another program running on the AS/400 system.

Starting a Session with a 3694 Document Processor

For a 3694 processor, the session and program are both started as a result of a program start request, which is an INIT-SELF request sent by the controller. On a secure AS/400 system, this INIT-SELF must also include information for determining a user ID and password.

Note: ICF finance supports only target programs for the 3694 processor. The 3694 must always establish the session to start the program on the AS/400 system. The AS/400 system cannot initiate a session on the controller.

If the 3694 processor sends optional data in the user data field, there must be a minimum of 10 bytes of data in the following format:

Figure 5-1. User Data Field Information

User Data Field	Description
1	Request code.
2 through 4	User ID; these 3 characters are added to USER to form the user identifier.
5 through 8	Password.
9 through 10	These 2 characters are added to LIB to form the library name. If these characters are 00, the current library list of the subsystem containing the communications entry is used.

The name of the program to be started is specified in the destination logical unit (DLU) field of the INIT-SELF command. For more information about the INIT-SELF command format, see Figure 7-2 on page 7-7.

Open/Acquire Operation

Communications sessions using ICF finance are always started with an acquire operation. Before any input or output operations can be directed to a program device, the program device must be acquired. Your application program uses the acquire operation to establish a session between your program and the finance controller. To be acquired, program devices must be defined to the ICF file by using one of the following commands:

- Add ICF Device Entry (ADDICFDEVE)
- Override ICF Device Entry (OVRICFDEVE)

A program device can be implicitly acquired through the open operation or explicitly acquired through the acquire operation. The acquire operation can be used many times with different program device names. When you explicitly acquire a device using the acquire operation, you identify the session you want to establish by using the same program device name on the acquire operation as you specified on the PGMDEV parameter for the ADDICFDEVE or the OVRICFDEVE command.

You must specify the RMTLOCNAME parameter on the Add or Override ICF Device Entry command. If the session is a controller-initiated session, you must specify *REQUESTER for the RMTLOCNAME parameter. If the session is an AS/400 system-initiated session, you must assign a value for the RMTLOCNAME parameter. The parameter must match the remote location name in the device description with which you want to communicate. For more information about starting your program, see “Starting Your Program” on page 5-3.

You can use the wait file (WAITFILE) parameter of the CRTICFF, OVRICFF, or CHGICFF command to specify the maximum amount of time you want to wait for session resources to become available.

Note: The WAITFILE value is not used for sessions where the connection to the controller is over a switched connection, for example, a SDLC switched line, a X.25 SVC, Ethernet, or a Token-Ring connection.

Sending Data

Data is sent to and received from the controller by using write, read, and write-with-read operations. The write operations are done using a record format. The results of these operations are sent to the program by using major and minor codes, high-level status values, and the input/output feedback area.

Finance communications buffers data. The first record and the last record of a group of records are sent to the controller prior to the completion of the output operation. However, output operations for records that are not the first or last in a group of records are buffered by finance communications. These records are sent when the buffer becomes full, or when the last record of the group is sent.

Write Operation

The write operation passes one or more data records from the AS/400 program to the remote controller program in this session. Finance communications determines the size of the record from the MAXLENRU parameter in the device description and will block your data into as many records as needed.

Note: Finance communications does not support multiple record groups for the 3694 processor.

Finance communications supports write operations while the session is invited. If the write operation ends successfully, and if the invite function is not specified on the write operation, the session is no longer invited when the write operation ends. If the write operation does not end successfully, or if the invite function was specified on the write operation, the session remains invited.

Force-Data Function

Your program uses the force-data function to send immediately (without waiting for the buffer to become full) any data currently being held in the buffer. Any data specified on the same operation as the force-data request is also sent. If data is not specified on a write operation with the force-data function, a null record is sent.

For more information about buffering data, see “Buffering” on page 7-1.

End-Of-Group Function

Use the end-of-group (ENDGRP) function to indicate the end of a user-defined group of records. When the ENDGRP function is issued, the control of the session goes back to a contention state. Then, either the AS/400 system program or the controller program can send data. When the ENDGRP function is used, finance communications requires a response from the finance controller to the group just sent. Control is not returned to the application program until a response is received.

Note: This function does not apply for a 3694 controller.

Function-Management-Header Function

Use the function-management-header function to send control information to the controller program about the data following the function-management-header. You must designate data on a write operation that specifies the function-management-header function. Your program is responsible for creating the function-management-header.

Receiving Data

You can use the read and read-from-invited-program-device operations to receive data. When your program receives data with a read or read-from-invited-program-devices operation, all the records of the group are given to your program. A read operation does not end until the entire group of records is received by the finance communications support.

For 47xx and FBSS controllers, the read operation also closes a partially sent group of records. If a group of records is implicitly closed by a read operation, a response from the controller program to the group of records being sent is not required.

Read Operation

Your program uses the read operation to obtain data from the controller. This operation causes finance communications to obtain data from the controller program with which your program is communicating. The read operation causes your program to wait for the data if it is not immediately available. Your program receives control when the data is available. The READ operation waits until a complete group of records is received. If an SNA Cancel command is received, the data is discarded and the read operation waits for the next complete group of records.

Finance communications always waits until an entire group of records is received before returning control to your program. If the length of the field in the record format you are using is large enough, the maximum data length that can be received is the value you specified in the MAXRCDLEN parameter of the Create ICF File (CRTICFF), Change ICF File (CHGICFF), and Override ICF File (OVRICFF) commands. If the format you are using is not large enough, you receive an error return code.

You do not need to issue an invite function before a read operation to receive data. However, if a program device has an outstanding invite to which a read is issued, the read completes the invite and receives the data.

Note: The read operation obtains data from a specified program device while the read-from-invited-program-devices allows data to come from any previously invited program device.

When a group of records is partially sent by your program, a read operation or a write-with-read operation that does not explicitly specify the end-of-group function implicitly sends the end-of-group function. When the group is closed implicitly, no response is required from the controller.

Note: A response is always required for a system monitor session, regardless of whether the group was implicitly or explicitly closed.

Invite Function

The invite function prepares your program to receive data. You must perform an output operation with the INVITE DDS keyword or with a system-supplied format to issue an invite function. You can combine additional output functions or data with the invite function. Your program can continue processing after issuing the invite request and does not need to wait for the data to arrive.

If a group of records was partially sent by your program, combining additional output functions or data with the invite function causes an implicit end-of-group function to be sent. If the end-of-group function is not explicitly specified (by using the end-of-group DDS keyword) with the invite function, a response to the group of records is not required.

Note: For a system monitor session, a response is required regardless of whether the end-of-group function was implicitly or explicitly specified.

Your program must issue a read-from-invited-program-devices operation or read operation to obtain the data.

Read-from-Invited-Program-Devices Operation

The **read-from-invited-program-devices operation** is an input operation that waits for input from any one of the invited program devices for a user-specified time. The read-from-invited-program-devices operation follows the invite function. After issuing an invite function, use the read-from-invited-program-devices operation to receive data from any previously invited program devices. The operation always ends when any of the invited program devices have received a complete group of records, or if the time limit set by the timer function has elapsed.

Waiting for a Display File, an ICF File, and a Data Queue

You can use data queues for a program that waits for data on a display file, an ICF file, and a data queue at the same time (in any combination). When you specify the DTAQ parameter for the following commands:

- Create Display File (CRTDSPF)
- Change Display File (CHGDSPF)
- Override Display File (OVRDSPF)
- Create ICF File (CRTICFF)
- Change ICF File (CHGICFF)
- Override ICF File (OVRICFF)

you can indicate a data queue that will have entries placed on it when either of the following occurs:

- An enabled command key or Enter key is pressed from an invited display device.
- Data becomes available from an invited ICF session.

By using the IBM-supplied QSNDDTAQ program, jobs running on the system can also place entries on the same data queue as the one specified in the DTAQ parameter.

For an ICF file or display file, the application program uses the IBM-supplied QRCVDTAQ program to receive each entry placed on the data queue and then processes the entry based on whether it was placed there by the display file, by an ICF file, or by the QSNDDTAQ program. For a display file, the application then issues a read or read-from-invited-program-devices operation to receive the data. For more information on the QRCVDTAQ function and syntax, and examples of waiting on one or more files and a data queue, see the *CL Programming* book.

The display file or ICF file entry that is put on the data queue is 80 characters in length and contains the field attributes described in Figure 5-2. Therefore, the data queue that is specified using the commands listed above must have a length of at least 80 characters.

Entries placed on the data queue by jobs using QSNDDTAQ are defined by the user.

Figure 5-2. Display File and ICF File Entry Field Attributes

Position	Data Type	Meaning
1 through 10	Character	The type of file that placed the entry on the data queue. This field can have one of two values: • *ICFF (ICF file) • *DSPF (display file) If the job receiving the data from the data queue has only one display file or one ICF file open, then this is the only field that needs to be used to determine what type of entry has been received from the data queue.
11 through 12	Binary	Unique identifier for the file. The value of the identifier is the same as the value in the open feedback area for the file. This field should be used by the program receiving the entry from the data queue only if more than one file with the same name is placing entries on the data queue.
13 through 22	Character	The name of the display or ICF file. This is the name of the file actually opened after all overrides have been processed and is the same as the file name found in the open feedback area for the file. This field should be used by the program receiving the entry from the data queue only if more than one display file or ICF file is placing entries on the data queue.
23 through 32	Character	The library where the file is located. This is the name of the library after all overrides have been processed and is the same as the library name found in the open feedback area for the file. This field should be used by the program receiving the entry from the data queue only if more than one display file or ICF file is placing entries on the data queue.
33 through 42	Character	The program device name after all overrides have been processed. This name is the same as that found in the program device definition list of the open feedback area. For file type *DSPF, this is the name of the display device where the command or Enter key was pressed. For file type *ICFF, this is the name of the program device where data is available. This field should be used by the program receiving the entry from the data queue only if the file that placed the entry on the data queue has more than one device or session invited prior to receiving the data queue entry.
43 through 80	Character	Reserved.

Notifying the Remote Program of Problems

Use the fail, cancel, and negative-response functions to inform the remote program about any errors in the data being sent or received.

Fail Function

Use the fail function to indicate an error has occurred when sending or receiving data.

If a program that is sending data issues a fail function, this indicates that errors exist in the data sent. A cancel indication is sent to the controller program.

You can use the fail function if your program receives data and finds an error in the received data. After finding the error, your program should issue the fail function as the next operation. A negative response with a sense code of

08110000 is sent to the controller from the AS/400 system.

Cancel Function

If you are sending data and decide to end a partially sent group of records, you can use the cancel function to cancel a group of records providing you have not sent the end-of-group indicator. Any buffered data is sent before the cancel request is sent.

The cancel function is not valid for the 3694 processor.

Negative-Response Function

If you are receiving data, use the negative-response function to inform the remote program that the data it sent contained an error.

Using a negative-response function is similar to issuing a fail function when receiving data, except that you can also include eight characters of sense data with the negative-response function. The sense data tells the sending system about the error in the data you received. The first two characters of the sense data must be either 10 or 08 or the first four characters must be 0000. All characters must be a value from 0 though 9, from A through F, or from a through f. For more information about sense data, see the *Systems Network Architecture Reference Summary* book.

The sense data accompanying the negative response goes to the normal output buffer. No other data can be sent with a negative-response function. If a sense code is not specified, a default sense code of 08110000 is used.

Using Additional Functions and Operations

The following functions give you additional control over the finance operations.

Cancel-Invite Function

If a program device is invited, you can use the cancel-invite function to cancel the invite function if data has not been received for the invited-program device. The cancel-invite function allows your program to cancel an invite operation, after which you can issue an input or output operation. If finance communications has already received data for the program device from the controller, the cancel-invite function is rejected with a 0412 return code.

Timer Function

Your program can use the timer function to set a timer and wait for it to end before performing some specified function. The timer function specifies an interval of time (in hours, minutes, and seconds) to wait before your program receives a timer-expired return code (0310).

Your program continues to run and all operations are valid during the time interval. To determine if the time limit set by the timer function elapsed, your program must use the read-from-invited-program-devices operation after issuing the timer function.

For more information about the timer function, see the *ICF Programming* book.

Get-Attributes Operation

You can use the get-attributes operation at any time after the file has been opened to determine the status of a program device. If the program device is not acquired, the information is obtained from the program device entry defined with the ADDICFDEVE or OVRICFDEVE command.

Ending a Session

The AS/400 application program must end the ICF finance session by using a release operation or an end-of-session function. Ending an ICF session does not necessarily affect the SNA session.

If the SNA session was established as a result of the remote system sending an INIT-SELF request, the SNA session may still be intact after your program ends the ICF finance session.

If the SNA session was established as a result of a source program issuing an ICF Acquire operation (an unsolicited start session request), the SNA session ends when the source program ends the ICF finance session.

The controller may request that the SNA session end by sending a TERM-SELF command, a Request Shutdown (RSHUTD) command, or an UNBIND command.

If no ICF session exists on the SNA session, you can end the SNA session by varying off the device using the VRYCFG command. For more information about SNA considerations, see “Systems Network Architecture (SNA)” on page 7-7.

Release Operation

For a source program, you can explicitly release a program device from an ICF file by using the release operation, or you can implicitly release the device by closing the file. If you release the program device, you must acquire it again before you can issue input/output operations. The release operation is accepted *only* if the program is not invited, if the last write operation contained an end-of-group indicator, and if no more data is waiting to be read.

For a target program, the release operation ends the logical connection between your program and the remote controller program. The program, or another program in the same job, can establish the connection again by acquiring the program device again. The communications session, including the state of the session, remains intact.

End-of-Session Function

The end-of-session function always ends the finance session. When your program issues an end-of-session (EOS) function, finance communications ends the program’s attachment to the session and frees the resources used during the session. Those resources are then made available to other programs waiting to establish a session. If you issue an end-of-session function when either your system or the controller program is sending or receiving data, finance communications still ends the session.

Note: The end-of-session function always ends the session if it exists. The only possible return codes from an end-of-session function are 0000 (normal end) or 830B (program device not acquired).

Close Operation

The processing done by the close operation depends on whether or not the file is shared. If the file is not shared, the file resources allocated by the open operation are deallocated and returned to the system.

If the file is shared, the program cannot do input/output operations on the file. Other programs that have the file open can still use the file.

If a close operation is successful, only an open operation is allowed to be used with the file. If the close operation fails, the program should issue the close operation a second time. A second close operation is always successful.

For a target program, the close operation ends the logical connection between your program and the remote controller program. The program or another program in the same job can establish the connection again by opening the file and acquiring the program device again. The communications session, including the state of the session, remains intact.

Using Response Indicators

Response indicators are specified in a 1-character field that is passed with an input record from the system to a program to provide information about the data record or actions taken by the work station user. To determine which response indicators to use when your program does a read operation, consider the following:

- What data was received from the remote program?
- What does the remote program expect of your program?
- What will be the next operation from your program?

Response indicators are only effective for input operations or a combined output, then input operation. They have no effect on an output operation. You can use multiple response indicators for a single input operation.

Receive End-of-Group

Use the receive end-of-group (RCVENDGRP) response indicator to determine if your program received a group of records. You receive one end-of-group record only once per group, and only one record contains the end-of-group indication. The major return code can be either 00, 02, or 03 and the minor return code is 03 or 07.

Receive Function-Management-Header

Use the receive function-management-header (RCVFMH) indicator to determine if your program received a function-management-header indicator. The major return code is 00 and the minor return code is 07 for the function-management-header indicator.

Receive Negative-Response

Use the receive negative-response (RCVNEGRSP) indicator to determine if an error exists in data that your program has sent. The major return code is 83 and the minor return code is 19 for the data sent by the RCVNEGRSP indicator.

Using Input/Output Feedback Area

The input/output (I/O) feedback area contains the results of read and write operations for your finance application programs as communicated by ICF return codes. This feedback area consists of two parts:

- A common input/output feedback area containing information affecting all communications types
- A file-dependent input/output feedback area containing information affecting one or more communications types

Offset 38, in the file-dependent part of the I/O feedback area, applies to finance. This field contains negative-response error data. For return code 8319, this area contains sense data indicating the reason the operation failed. For more information about sense codes, see the book *SNA Formats*.

For more information about the input/output feedback area, see the *ICF Programming* book.

Using Return Codes

After an operation ends, a return code (and a high-level language file status) is sent to your application. This return code indicates if your operation ended successfully. Exception messages can also be issued along with the return code. For more information about return codes, see Appendix B, “Return Codes, Messages, and Sense Codes.” For more information about the high-level file status, see the appropriate language reference book.

Note: ILE C/400 language does not use high-level file status. However, your ILE C/400 program can use the `errno.h` statement to check for any I/O exceptions that may occur and to retrieve any exception data associated with the error. Your program should check the return code to determine the following:

- Status of operation just ended
- Operation to be done next

For example, when issuing a read operation, a major return code of 00 on an input operation indicates that the operation ended successfully. Along with this major code, you can also receive one of the following minor return codes from finance communications:

01 Indicates your program completed a successful invite function.

03 Indicates that your program has received a group of records.

07 Indicates that a function-management-header was received by your program in this group of records.

Another example is a major code of 83, which indicates an error was found that could be recoverable. Different minor codes can be returned with this major code, just as with major return code 00. For example, if your program receives a return code of 831C, this means your program failed to correctly process a return code to a previous write operation. Data was available to be received. To correct the problem, issue an input operation to receive the data.

Your program should check ICF return codes when every operation ends to be certain that the operation completes successfully or that appropriate recovery action was taken.

For a description of return codes that can be returned to your finance application, see Appendix B, “Return Codes, Messages, and Sense Codes.”

Chapter 6. Writing Non-Intersystem Communications Function Application Programs

The information in this chapter is intended to help the customer write finance application programs using the non-ICF interface. It contains general-use programming interfaces, which allow the customer to write programs that use the services of the OS/400 program.

This chapter describes starting a non-ICF finance application program using the Submit Finance Job (SBMFNCJOB) command. It also discusses using finance input/output manager (FIOM) and user-defined data stream (UDDS).

Using non-ICF finance represents three levels of support, with the SBMFNCJOB command providing the most support. Your application programs are not required to do any communications operations, since non-ICF finance communicates through a display file using UDDS. UDDS gives your program total control of the data stream. Using FIOM allows your program to communicate with finance devices by calling input/output (I/O) routines rather than doing communications operations.

Figure 6-1 shows the input data format for non-ICF finance. The incoming data contains a 3-byte header. The outgoing data format for non-ICF finance must contain UDDS control bytes and a 4700 finance control byte, as shown in Figure 6-2.

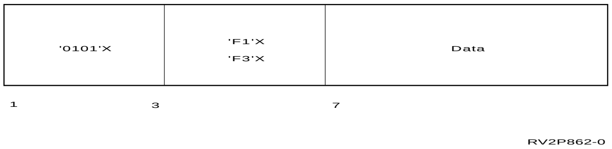
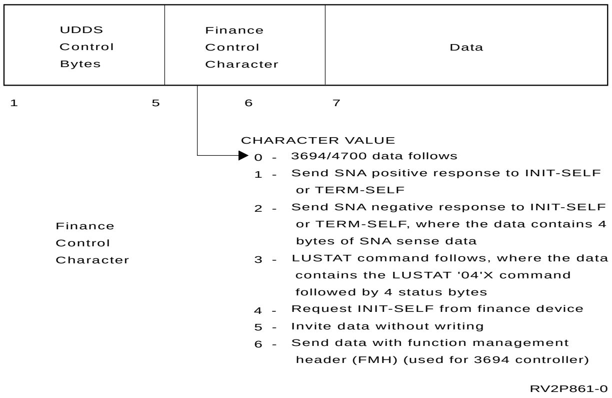


Figure 6-1. Incoming Data Control Bytes



0 - 3694/4700 data follows
1 - Send SNA positive response to INIT-SELF or TERM-SELF
2 - Send SNA negative response to INIT-SELF or TERM-SELF, where the data contains 4 bytes of SNA sense data
3 - LUSTAT command follows, where the data contains the LUSTAT '04'X command followed by 4 status bytes
4 - Request INIT-SELF from finance device
5 - Invite data without writing
6 - Send data with function management header (FMH) (used for 3694 controller)

Figure 6-2. Outgoing Data Control Bytes

Figure 6-3 shows the operations you can do with each method without directly specifying the operation in your program.

Figure 6-3. Operation Capabilities for Non-ICF Finance

Operation	SBMFNCJOB	FIOM	UDDS
Open file	Yes	No	No
Acquire devices	Yes	No	No
Handle INIT-SELF and TERM-SELF requests	Yes	No	No
Build and format data stream	Yes	Yes	No
Release devices	Yes	No	No
Close file	Yes	No	No
Error recovery	Yes	No	No

For information concerning programming, security considerations and procedures for non-ICF communications, see Chapter 7, "Finance Considerations."

Using the Submit Finance Job Command

This section describes the SBMFNCJOB command, including how and when to use it. Processing examples and information describing how to set up a device, user, and program tables are included. These tables are used by the SBMFNCJOB command. Error handling is also discussed. For more information and a syntax diagram of the SBMFNCJOB command, see the *CL Reference* book.

A finance job started with the SBMFNCJOB command is an interface between the AS/400 finance application programs and the 4701 or 4702 controller application programs.

Use the SBMFNCJOB command if all of the following situations exist:

- The AS/400 system communicates with a 4701 or 4702 finance controller.
- A finance device table and a finance program table have been defined (defining a finance user table is optional) using the WRKDEVTBL and WRKPGMTBL commands as described in “Work with Device Table Command” on page 3-5 and in “Work with Program Table Command” on page 3-7.
- The device types 3277, 3278, or 3279 are not entered in your device table.
- The USER parameter supplied on the job description, under which the finance job runs, is USER (QFNC). Specify this parameter using either the Create Job Description (CRTJOBD) or Change Job Description (CHGJOBD) command. For more information about these commands, see the *CL Reference* book.
- The 4701 or 4702 controller application program sends data, or transactions, first and expects to receive data back.
- The 4701 or 4702 controller application program passes data in the proper format, as described in “Processing Transactions” on page 6-6.

The SBMFNCJOB command submits a batch job to the QFNC subsystem through the QFNC job queue. This batch job does the following:

- Acquires the devices in the device table.
- Invites these devices to allow data to be received from them.
- Verifies that the user ID received with the INIT-SELF request matches an entry in the finance user table. This applies only if a user table was created and its name was specified in the SBMFNCJOB command.
- Calls the program requested by the finance controller to process the transaction if the program is specified in the program table.
- Returns data formatted by your AS/400 application to the finance controller.
- Releases the device when your finance controller requests a session end.

Parameters

When using the SBMFNCJOB command, specify the following parameters:

DEVTBL
Specifies the name of the device table that the finance job uses to determine which 4704 or 3624 devices the finance job controls. This is a required parameter.

PGMTBL
Specifies the name of the program table the finance job uses to determine, from the program ID (sent in the data stream with a finance transaction), which system user program names process the finance transaction. This is a required parameter.

USRTBL
Specifies the name of the user table that the finance job uses to verify a finance user when a finance sign-on is received.

JOB NAME
Specifies the job name that is associated with the submitted finance job. The possible values are:

- QFNCJOB: The job is submitted as QFNCJOB.
- Job_name. Specify the user-defined job name that is associated with the submitted finance job.

JOBD
Specifies the name of the job description the finance job uses.

MSGQ
Specifies the name of the message queue, if any, to which messages are sent while the finance job is running.

Error Handling

This section describes error handling support for a finance job started with the SBMFNCJOB command.

Input/Output: Finance support attempts error recovery whenever a finance job receives an I/O exception. When an I/O exception signals a finance job, the major/minor return code is retrieved from the message to determine the potential of recovery from the error. Recoverable errors alert the finance job to try recovery. If successful, processing continues normally. If unsuccessful, further action depends on the nature of the error.

Device errors result in the release of the affected device and continued processing of other devices associated with the job. However, if recovery is not successful, the controller or line errors end the job.

When a permanent I/O error is received, that error is handled the same as an unsuccessful recovery. Device failures cause the release of the affected device but communication with remaining devices is continued. If other I/O exceptions are too severe to be handled within the job, the job ends.

Non-Input/Output: For a finance job, the SBMFNCJOB command handles non-I/O errors as most AS/400 jobs are handled. If an error occurs, a message is sent to the message queue associated with the finance job. The SBMFNCJOB command also allows you to specify the name of the message queue where you want certain informational messages sent. Messages sent to this finance queue relate to general error conditions that happen when the finance job runs.

Some of the informational messages include:

```
CPI8372 - ERRORS OCCURRED DURING THE START OF FINANCE JOB

CPI8379 - FINANCE TRANSACTION FAILED

CPI8380 - FINANCE DEVICE FAILED

CPI8394 - PROGRAM ID NOT IN PROGRAM TABLE
```

This additional message queue allows you to supervise the status of your finance jobs more effectively.

For additional information about AS/400 finance support messages, see the online message help information.

Supervising Finance Jobs

To supervise finance jobs, you can do the following:

- Specify MSGQ (*NONE) on the SBMFNCJOB command when you submit a job. Then, if an error occurs when starting finance support, a message appears in the message queue.
- Use the Work with Configuration Status (WRKCFGSTS) command to ensure that all devices allocated to a finance job have been successfully acquired.
- Use the Work with Job Queue (WRKJOBQ) command to display any finance jobs that have been submitted and are waiting to run when the QFNC finance controller is inactive:

  ```
  WRKJOBQ JOBQ(QGPL/QFNC)
  ```

- Use the Work with Active Job (WRKACTJOB) command to display and change the status and performance information for submitted finance jobs:

  ```
  WRKACTJOB SBS(QFNC)
  ```

- Use the Display Job Log (DSPJOBLOG) command, with the job names as parameters, to display messages held in the job log when running your finance jobs.

  For an explanation of the format and presentation of the IBM-supplied messages and the complete finance support message texts, see the online messages.

For more information about these commands, see the *CL Reference* book and *Work Management* book.

The SBMFNCJOB command has a JOB parameter that makes supervising and canceling finance jobs easier. Use this parameter to supply unique names for your finance jobs so that you can easily distinguish them from other jobs in your system.

Data Flow Examples

The following figures show data flow examples when using the SBMFNCJOB command. Figure 6-4 on page 6-4 shows a finance job verifying that the user ID received from the controller application exists in the user table associated with that job.

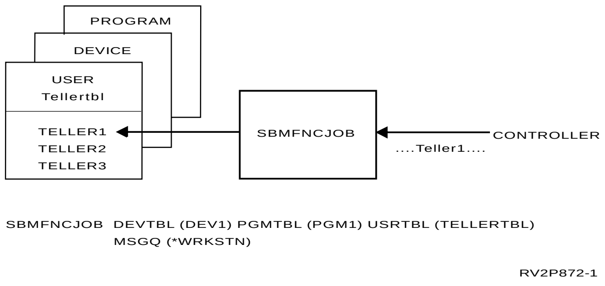
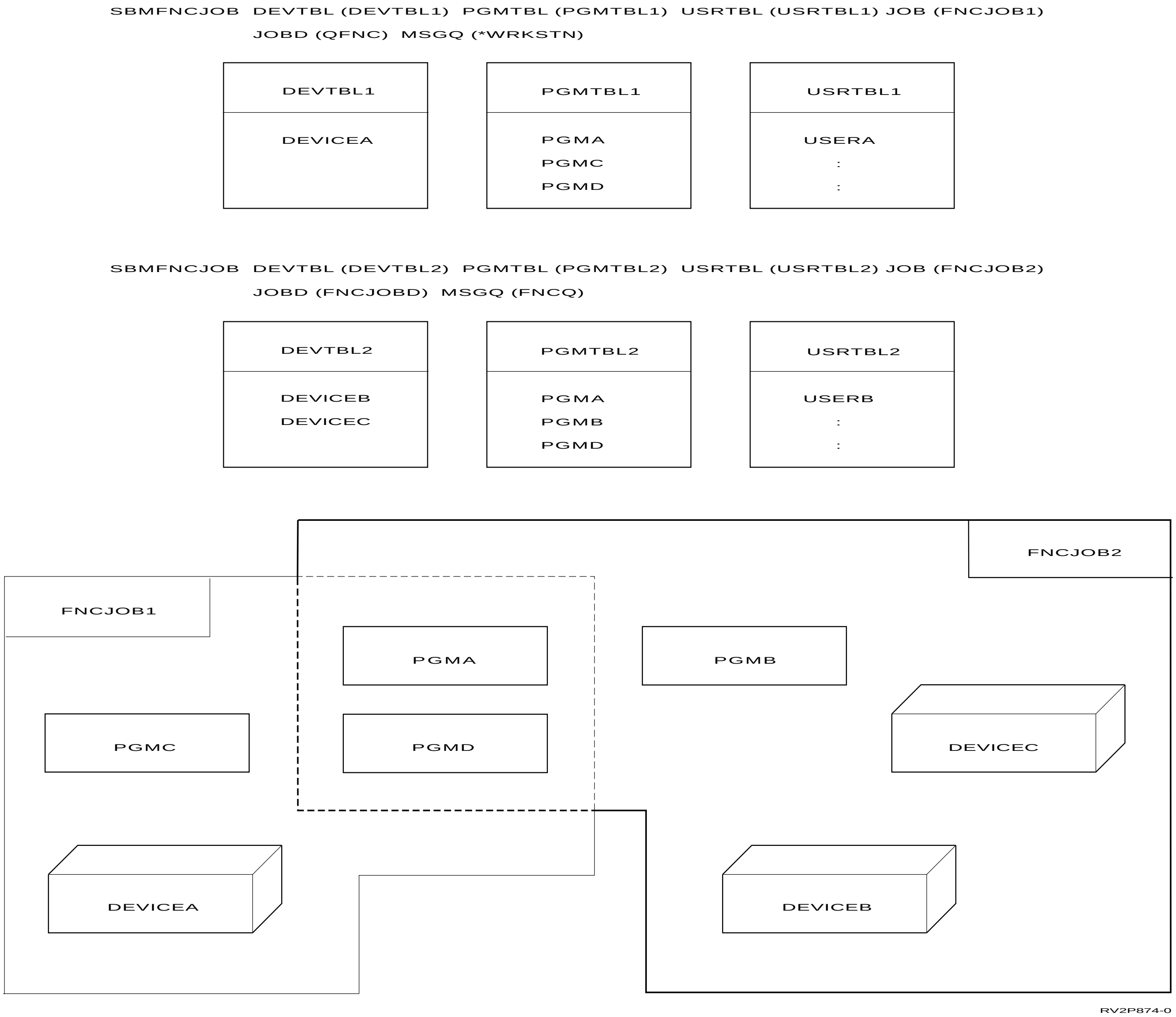


Figure 6-4. INIT-SELF Request Approved by the User Table

Figure 6-4 shows how a user ID sent in the INIT-SELF data stream is validated using the user table named on the SBMFNCJOB command parameter.

Figure 6-5 on page 6-5 represents an environment in which more than one finance job is submitted. The figure shows that two finance jobs control the finance devices. The finance jobs share certain application programs but cannot share devices.

Notice in Figure 6-5 that DEVICEA is acquired by FNCJOB1 and devices DEVICEB and DEVICEC are acquired by FNCJOB2. The two finance jobs cannot share devices. Even if only one device name is specified in two device tables, only one finance job can acquire that device. However, finance jobs can share application programs. The two finance jobs share programs PGMA and PGMD in Figure 6-5 on page 6-5.



```
SBMFNCJOB  DEVTBL (DEVTBL1)  PGMTBL (PGMTBL1)  USRTBL (USRTBL1) JOB (FNCJOB1)
           JOBD (QFNC)  MSGQ (*WRKSTN)
```

DEVTBL1	PGMTBL1	USRTBL1
DEVICEA	PGMA	USERA
	PGMC	:
	PGMD	:

```
SBMFNCJOB  DEVTBL (DEVTBL2)  PGMTBL (PGMTBL2)  USRTBL (USRTBL2) JOB (FNCJOB2)
           JOBD (FNCJOBD)  MSGQ (FNCQ)
```

DEVTBL2	PGMTBL2	USRTBL2
DEVICEB	PGMA	USERB
DEVICEC	PGMB	:
	PGMD	:

Figure 6-5. Two Finance Jobs Control the Finance Devices

Figure 6-6 shows how a transaction sent in a data stream from the controller application, with the finance job calling the requested transaction processing program, can be handled.

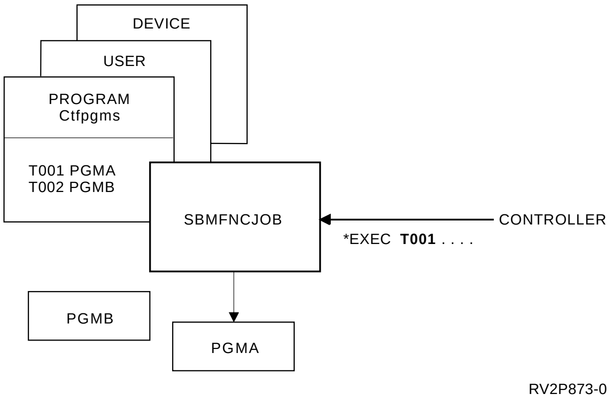


Figure 6-6. Controller Processing Transaction T001

In Figure 6-6, the controller requests processing of transaction T001. The SBMFNCJOB interface determines from the program table that the AS/400 system application PGMA should process the transaction.

Processing Transactions

By using the SBMFNCJOB command, financial transactions can be processed from the 4701 or 4702 controller on the AS/400 system, or from the AS/400 system on the 4701 or 4702 controller. This topic describes how each process occurs.

Sending Data from the Finance Controller to the AS/400 System: When you use the SBMFNCJOB command, the finance controller application program must use one of the data stream formats shown in Figure 6-7.

Figure 6-7 shows two data streams being sent from the finance controller. These data streams represent the two formats accepted by the finance job. The first data stream format contains the characters *EXEC, followed by a single blank, a program ID, another blank, and data. The second format has no *EXEC characters; this data stream begins with the program ID, which is again followed by a single blank and data.

When you use either data format, the program ID (PGM ID) can be of variable length from 1 to 8 characters and must be followed by a single blank. A maximum of 512 bytes of data is allowed per transaction.

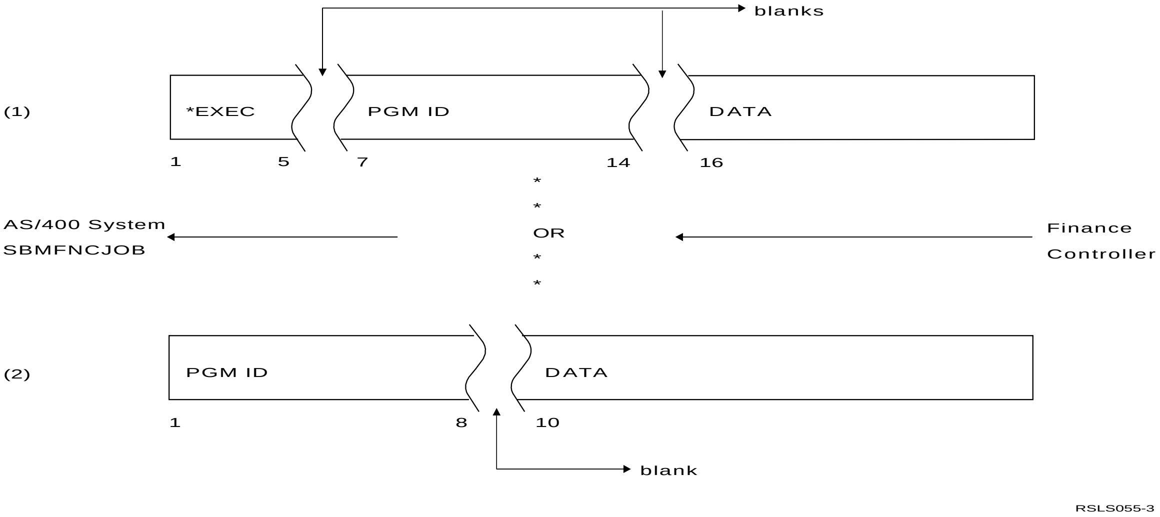


Figure 6-7. Format of Data Streams When Using the SBMFNCJOB Command

When the finance job receives the data stream from the finance controller, it uses the finance program table you created to determine which of your AS/400 application programs to call. Your AS/400 application program is passed through the following SBMFNCJOB parameters:

device name
A character variable, 10 bytes in length, representing the name of the 3624 or 4704 device sending the data.

data length
A decimal field, 15 bytes in length with 5 decimal positions, containing the length of the data received.

data
A character variable, 512 bytes in length, containing the data received from the finance controller. The data length parameter determines the actual length of the finance data in the data parameter.

Figure 6-8 shows the expected parameters.

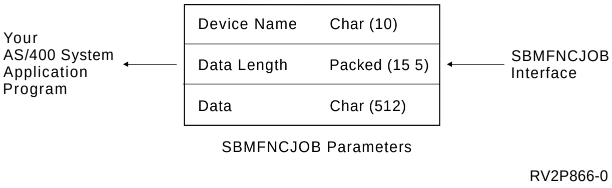


Figure 6-8. Parameters Expected by the AS/400 Application Program

Note: The maximum size of a character variable (referred to in Figure 6-8) might be restricted to fewer than 512 bytes by the high-level language in which your application programs are written. If so, see the programmer's guide for the programming language you are using, or the user's guide for the controller application to ensure that the controller does not send more data than your AS/400 application program can receive.

Sending Data from the AS/400 System to the Finance Controller:

To send data from your AS/400 application program back to the finance controller application program:

1. Set the data length parameter to the length of the data you want to send.
2. Move your new data into the data parameter.

If you do not want to send data back to the finance controller application program, set the data length parameter to 0 bytes.

Using Finance Input/Output Manager

This section describes the finance input/output manager (FIOM) call, including how and when you can use it. Processing specifications and error handling are also discussed.

FIOM is a flexible tool for use in a variety of finance environments. FIOM supplies a high-level connection to UDDS communications for AS/400 finance support users. It simplifies logical unit 0 (LU0) communications between your AS/400 transaction application program and the controller application program. FIOM is an alternative to transparent UDDS communications when you want direct conversation between your AS/400 transaction-processing application program and the finance controller application.

The FIOM support formats, sends, and receives UDDS. Figure 6-9 and Figure 6-10 on page 6-8 compare communications using UDDS and FIOM. In transparent UDDS communications, shown in Figure 6-9, your AS/400 application program communicates with the controller application using SNA LU0 protocols. Data streams must be framed in UDDS control information.

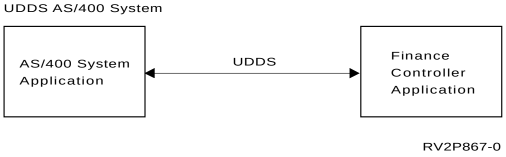


Figure 6-9. Transparent UDDS Communications

FIOM makes this communication easier as shown in Figure 6-10 on page 6-8. Information you supply to the FIOM routines allows for UDDS communication, yet removes many of the requirements for formatting and sending UDDS in your high-level application.

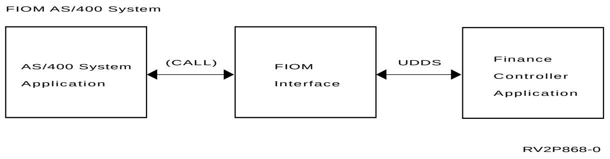


Figure 6-10. UDDS Communications with FIOM

Finance Input/Output Manager Routines

FIOM is called by your program. **FIOM** consists of four routines that can be called as external subroutines by your application programs to do I/O operations on a finance device that is configured as a non-ICF device. Each of the four routines accepts the parameters you supply and then performs the input/output (I/O) functions.

The following discussions supply a description of each routine. Each of the four FIOM routines discussed has an associated DATA parameter.

The DATA parameter for the following four routines allows a 512-byte field. However, an ILE RPG/400 program sets the maximum length of a character field at a value of 256 bytes. To determine the maximum length of data that can actually be sent for your application programs, see the appropriate programmer's guide.

QFN-Write Routine: The QFN-write (QFNWRT) routine accepts data from your AS/400 application program and sends the formatted data to the finance device you specify.

The following formats for the input parameters are passed to the QFNWRT routine:

Device ID
A 10-byte character field specifying the name of the finance device to receive the data.

Data length
A decimal field containing the data length to be written. The data length is defined as 15 bytes with 5 decimal positions. The maximum data length is 512 bytes.

Data
A character field containing data to be written to the device. The format of the data depends on the value assigned to the data type parameter. If the value of the data type is 0, the format of the data depends on protocols established for communications between your AS/400 application and the controller application program. If the value of the data type is 1, QFNWRT ignores the data parameter because the data stream being sent to the controller application has a specific, predetermined format. Other data type values, 2 and 6, correspond to data stream formats that you define to satisfy both SNA and AS/400 application program requirements.

Data type
A 1-byte character field that contains the type of data (4700 control character) written to the finance device. Valid data types are:

0 3694, 4701, and 4702

1 Positive response

2 Negative response, with the data parameter containing a minimum of 2 bytes of status code followed by sense data

6 Send 3694 data type with function-management-header

The following example is a call to the QFNWRT routine from an ILE RPG/400 program:

```
CALL 'QFNWRT'
PARM          WSID    10
PARM          SNDLEN  155
PARM          DATA    256
PARM          WRTTYP  1
```

QFN-Write/Invite Routine: The QFN-write/invite (QFNWRTI) routine works in the same manner as the QFNWRT routine, and also invites a response from the finance device.

The following formats of the input parameters are passed to the QFNWRTI routine:

Device ID
A 10-byte character field specifying the name of the finance device to receive the data and to be invited for communication.

Data length
A decimal field containing the length of the data to be written. The data length is defined as 15 bytes with 5 decimal positions. The maximum data length is 512 bytes.

Data
A character field containing data to be written to the device. The format of this data

depends on the value assigned to the data type parameter. If the value of the data type is 0, the format of the data depends on protocols established for communications between your AS/400 application and the controller application program. If the value of the data type is 4 or 5, the data parameter is ignored by the QFNWRTI routine because the data stream being sent to the controller application has a specific, predetermined format. Other data type values, 3 and 6, correspond to data stream formats that you define to satisfy both SNA and AS/400 application program requirements.

Data type
A 1-byte character field containing the type of data (4700 control character) written to the finance device. Valid data types are:

0	3694, 4701, and 4702
3	LUSTAT command follows, with the data parameter containing the LUSTAT X'04' command, followed by a minimum of 2 bytes of status
4	Request INIT-SELF command from finance device
5	Invite the finance device without writing data to the device
6	Send 3694 data type with the function-management-header

This is an example of a call to a QFNWRTI routine from an ILE RPG/400 program:

```
CALL 'QFNWRTI'
PARM            WSID    10
PARM            SNDLEN  155
PARM            DATA    256
PARM            WRTTYP  1
```

QFN-Read Routine: The QFN-read (QFNREAD) routine accepts data sent from a specific finance display in response to an invite operation sent by the QFNWRTI routine.

The following parameter formats are passed to the QFNREAD routine:

Device ID
A 10-byte character input field specifying the name of the finance device from which to read.

Data length
A decimal output field containing the length of the data read. The data length is defined as 15 bytes with 5 decimal positions. The maximum data length is 512 bytes.

Note: The data length parameter must be started by your application program and must contain a valid packed decimal (15 5) number before the QFNREAD routine is called.

Data
A character output field containing the data read from the finance device. The format of this data depends on the parameter value of the data type. If the value of the data type is 1, the format of the data depends on protocols established for communications between your AS/400 application and the controller application program. If the value of data type is 3, the data stream returned to the AS/400 application has a specific, predetermined format.

Note: The data parameter must represent a field in your application program that is large enough to contain the expected input data. If the receiving field is not large enough, adjacent data space can be overwritten with financial data.

Data type
A 1-byte output character field containing the type of data (4700 control character) read from the finance device. Valid data types are:

1	3694, 4701, and 4702 (no function-management-header)
3	Function-management-header and data to follow (INIT-SELF, TERM-SELF, or 3694, 4701, and 4702 data)

This is an example of a call to a QFNREAD routine from an ILE RPG/400 program:

```
CALL 'QFNREAD'
PARM            WSID    10
PARM            RCVLEN  155
PARM            DATA    256
PARM            RDTYP   1
```

QFN-Read/Invited Routine: The QFN-read/invited (QFNREADI) routine accepts input from any one of the invited finance devices associated with the finance job, returning the data to your application along with the name of the device from which it was received.

The following parameter formats are passed to the QFNREADI routine:

Device ID
A 10-byte output character field that specifies the name of the finance device from which data was read.

Data length
A decimal output field containing the length of the data that was read. The data length is defined as 15 bytes with 5 decimal positions. The maximum data length is 512 bytes.

Note: The data length parameter must be started by your application program and must contain a valid packed decimal (15 5) number before QFNREADI is called.

Data
A character output field containing the data read from the finance device. The format of this data depends on the value of the data type parameter. If the value of the data type is 1, the format of the data depends on protocols established for communications between your AS/400 application and the controller application program. If the value of the data type is 3, the data stream returned to the AS/400 application has a specific, predetermined format.

Note: The data parameter must represent a field in your application program that is large enough to contain the expected input data. If the receiving field is not large enough, adjacent data space can be overwritten with financial data.

Data type
A 1-byte character output field that contains the type of data (4700 control character) read from the finance device. Valid data types are:

1 3694, 4701, and 4702 data

3 Function-management-header and data to follow (INIT-SELF, TERM-SELF, or 3694, 4701, and 4702 data)

This is an example of a call to a QFNREADI routine from an RPG/400 program:

```
CALL 'QFNREADI'
PARM            WSID    10
PARM            RCVLEN  155
PARM            DATA    256
PARM            RDTYP   1
```

Error Handling

Any errors received by the FIOM routines during transaction processing produce diagnostic messages describing the errors. The CPF8390 escape message appears after these messages. For example, if the FIOM routines receive an I/O error, the diagnostic message CPD8289 (I/O errors occurred) is signaled and the escape message CPF8390 (errors occurred when running the program) is sent. Figure 6-11 on page 6-11 shows a list of the message identifiers sent by the FIOM routines.

The ability of your application program to handle error conditions depends on the language in which the program is written. For specific error-handling capabilities and error-recovery procedures for ICF finance, see the programmer’s guide for the language.

Figure 6-11. FIOM Error Messages

Message ID	Message Description	QFNWRT[1]	QFNWRTI[1]	QFNREAD[1]	QFNREADI[1]
CPD8280	Device not found	X	X	X	
CPD8281	Not authorized to device	X	X	X	
CPD8284	Invalid data length	X	X		
CPD8286	Invalid format of data length parameter	X	X	X	X
CPD8287	Invalid data type	X	X		
CPD8289	I/O errors occurred	X	X	X	X
CPD8290	OPEN errors occurred	X	X	X	X
CPD8291	CLOSE errors occurred	X	X	X	X
CPD8384	Unable to validate device description	X	X	X	
CPF8390	Errors occurred during program processing	X	X	X	X

[1] Call producing the messages.

If the programming language has error-handling capabilities, the application program can attempt recovery after receiving an I/O error from FIOM. I/O, OPEN, and CLOSE messages supply the major and minor return codes of the operation as part of the message replacement text. If the application program can retrieve this information, then you can try error recovery.

If the AS/400 finance job receives I/O errors during processing, and if the finance controller indicates host system format errors, examine the format of data streams being sent by the system application. These exceptions usually indicate that the data is not formatted correctly.

Using Finance Input/Output Manager with the Submit Finance Job Command

FIOM can extend the communications capabilities of your AS/400 finance application programs when the finance job is used as the primary interface to the finance controller application.

As discussed in “Processing Transactions” on page 6-6, when a finance job receives a transaction, one of your application programs is called to process the transaction. With regard to that application, you can expect the following to occur to a finance job submitted through the SBMFNCJOB command:

- Your application receives the finance device name, data length, and data as input parameters.
- Your application returns the data length and data parameters as output.
- When control returns to the finance job, the device named as input to your application remains in the same status as it was prior to the start of your program. Specifically, the following must occur:
  - The device must be acquired.
  - The device must have an active session. Therefore, if a TERM-SELF request is received by your application program, you must either send a negative response or process the TERM-SELF request and not return control to the finance job until another INIT-SELF request is received, and a session is active again with the device.
  - The device must be ready to be invited for further communications. The finance device is invited when the data you return from your program is written to that device.
- When control returns to the finance job, all devices, except the device named as input to your application, exist (acquired or unacquired; invited or uninvited; session active or inactive) as they did before your application program was called. For example, assume

that within your application, the QFNREADI routine is called to receive data from any invited finance device. After data is received from a device, call the QFNWRTI routine to invite the device again. In this way, when control returns to the SBMFNCJOB interface, the device is invited as it was before your application program was called.

If the preceding conditions do not occur, when control returns to the finance job, the results cannot be predicted. If the conditions do occur, the SBMFNCJOB interface allows independent communications between your application program and the finance controller application. After a finance job starts your program, and if the system exists as described in the previous paragraphs, your program can communicate directly with the controller application when control is returned to the finance job. Figure 6-12 on page 6-13 is a diagram of this communication.

Figure 6-12 on page 6-13 shows an environment in which a finance job is active on the AS/400 system using the OS/400 licensed program. When the finance job receives a transaction, one of your application programs is called to process the transaction. In turn, your program can call the four FIOM routines to communicate with the device. When the communications and all other transaction processing completes, your program should return control to the finance job, keeping the conditions previously described. The finance job returns the data supplied as output from your application program to the finance device sending the transaction. The device is invited again, and the finance job is ready to receive another transaction.

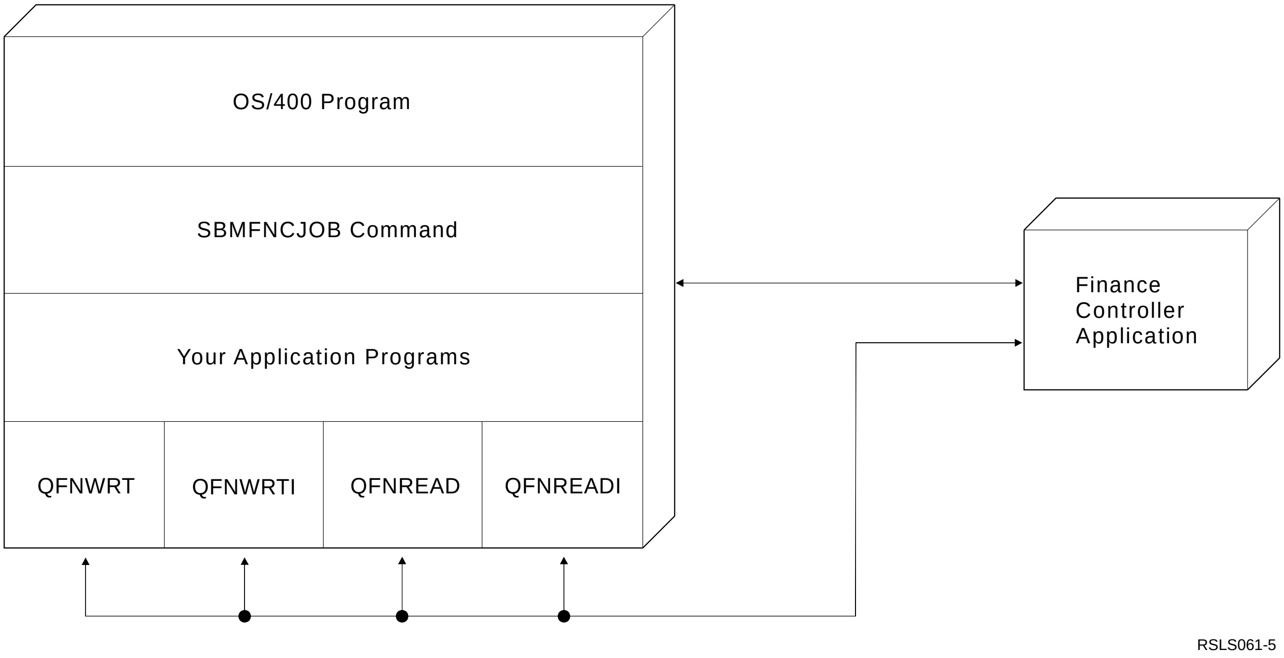


Figure 6-12. Communications between the Application Program and the Finance Controller

An advantage gained from using FIOM in a finance job environment is having more than 512 bytes of data associated with a particular transaction. If, for example, your ILE RPG/400 application program must write 800 bytes of data to complete a transaction, the data stream can be made into segments of 256, 256, 256, and 32 bytes. The 256-byte segments can be returned to the device through calls to the QFNWRT routine, with the last 32 bytes of data returned through the finance job. While this capability exists through UDDS, using FIOM is an easier method.

Using Finance Input/Output Manager without the Submit Finance Job Command

You can use FIOM without the SBMFNCJOB routine to simplify communications between your application programs and the controller application program. In environments where communications is routed by another method other than the SBMFNCJOB command, the four FIOM routines simplify communications for the routing program itself.

Figure 6-13 on page 6-14 shows how to use FIOM to handle communications for both the communications router and the application programs.

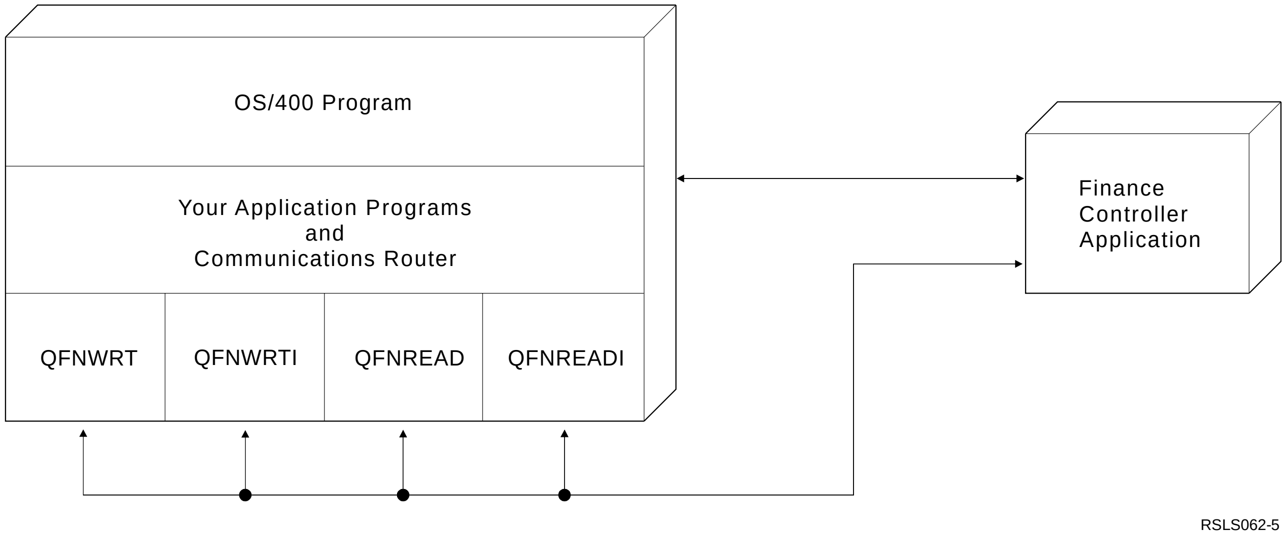

OS/400 Program
Your Application Programs and Communications Router
QFNWRT
QFNWRTI
QFNREAD
QFNREADI
Finance Controller Application
RSLS062-5

Figure 6-13. Communications for the Communications Router and Application Programs

This diagram shows an environment in which the finance job is not used on the AS/400 system. Your programs are responsible for routing the finance communications and for processing all data streams received from the finance controller. Since your programs are communicating directly with the controller, use the FIOM routines to simplify this communications by handling the read, write, and invite operations when called by your AS/400 application.

Your program is also responsible for acquiring and releasing the devices. To handle UDDS, you must compile your program against an externally described display file named QDFNDATA. This display file contains the following record formats:

```
R UDDSDTA1                    KEEP
  DATA       518    B   1  2
R UDDSDTA2                    INVITE
  DATA       518    B   1  2
```

This file must be placed before the QSYS library in your library list. After compiling the program, remove the library containing the file from the library list. Then your application program uses QDFNDATA file in QSYS library.

Notice in Figure 6-13 that the direct communications path still exists between the router and controller applications. This indicates additional communications occurrences, such as your router opening and closing the QDFNDATA display file or acquiring and releasing finance devices. While these tasks must be done independently of FIOM, a similar principle can be used to handle the communications; you can write your own subroutines to open and close the file and to acquire and release devices.

Using User-Defined Data Stream

This section discusses formatting information for UDDS and contains examples of communicating using UDDS.

Rather than using finance support, you can use UDDS to control and process the data streams. You must define a display file with record formats containing the user-defined (USRDFN) keyword. Then you can perform the usual input and output operations on the device by using these record formats.

Specify the USRDFN keyword at a record level (fields are not allowed on formats) by using the following steps.

1. Define an externally described display file and create your program using the record formats in this file. The file must not have record formats with the USRDFN keyword. Note the following example:

```
R  UDDS1                    KEEP
   DATA    518   B   1   2
R  UDDS2                    INVITE
   DATA    518   B   1   2
```

2. Create a second file with the same name as the first file. When you create this file, you must specify LVLCHK(*NO) in the Create Display (CRTDSPF) command.

The record format in this file must contain the USRDFN keyword. Note the following example:

```
R  UDDS1                   USRDFN
                           KEEP
R  UDDS2                   USRDFN
                           INVITE
R  DATAREC
   DATA    518       1   2
```

3. When running your program, use the second display file.

The INIT-SELF and TERM-SELF requests are sent to the AS/400 system by the controller. The AS/400 application program must do the following:

- Respond to the INIT-SELF and TERM-SELF commands.
- Process the transaction requested by the finance terminal operator.
- Send a write instruction to communicate with the finance terminals.
- Process the data stream associated with the write instruction in the UDDS format.

Formats

This section provides examples of the format that UDDS control bytes must follow.

Control Bytes: The finance control bytes for UDDS are shown in Figure 6-14 and Figure 6-15.

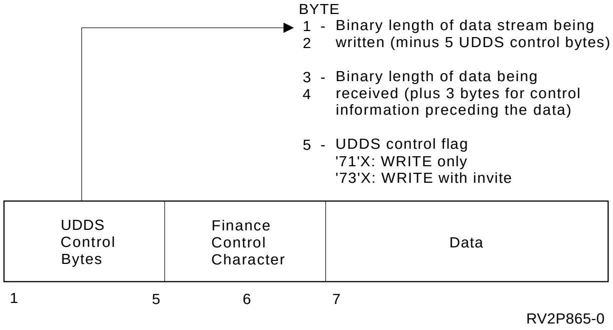

BYTE
1 - Binary length of data stream being
2 written (minus 5 UDDS control bytes)
3 - Binary length of data being
4 received (plus 3 bytes for control information preceding the data)
5 - UDDS control flag
'71'X: WRITE only
'73'X: WRITE with invite
UDDS Control Bytes
Finance Control Character
Data
1
5
6
7
RV2P865-0

Figure 6-14. UDDS Control Bytes

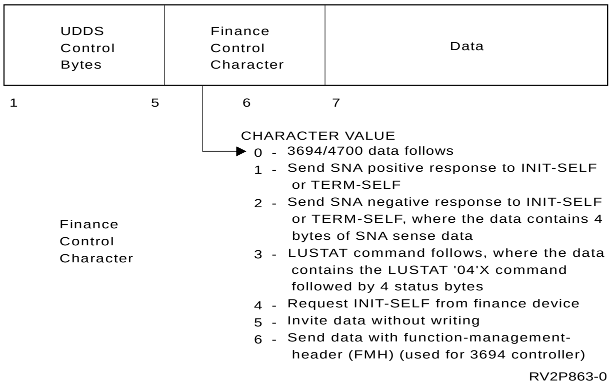

UDDS Control Bytes
Finance Control Character
Data
1
5
6
7
CHARACTER VALUE
0 - 3694/4700 data follows
1 - Send SNA positive response to INIT-SELF or TERM-SELF
2 - Send SNA negative response to INIT-SELF or TERM-SELF, where the data contains 4 bytes of SNA sense data
3 - LUSTAT command follows, where the data contains the LUSTAT '04'X command followed by 4 status bytes
4 - Request INIT-SELF from finance device
5 - Invite data without writing
6 - Send data with function-management-header (FMH) (used for 3694 controller)
Finance Control Character
RV2P863-0

Figure 6-15. UDDS Finance Control Character

Data coming from the 3694, 4701, or 4702 controller is shown in Figure 6-16. The data field follows three bytes of control information.

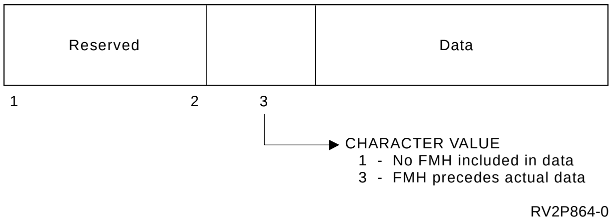


Figure 6-16. Format of Data from a 3694, 4701, or 4702 Controller

The AS/400 application program you use to communicate with a controller (using UDDS) must do the following:

- Acquire the finance devices.
- Identify and respond to the INIT-SELF requests.

  The application program issues a write/read operation using UDDS to receive the INIT-SELF request.
- Set the 4700 control character.
- Verify the finance terminal operator IDs and restrict their use, if necessary.

  When you receive an INIT-SELF request, your application program must send either an SNA positive or negative response and the correct 4700 control character.
- Call your transaction processing program.

  After your AS/400 application program sends the positive or negative response to the INIT-SELF, the application program can either write data to the controller (4700 control character set to 0), or it can invite the controller to send data by issuing a write/read with the 4700 control character set to 5.

- Release the finance device when you receive a TERM-SELF request.

  When your AS/400 application program receives the TERM-SELF, it must issue a write operation to send a positive response (4700 control character set to 1) and then release the device, or send a negative response (4700 control character set to a value of 2 bytes) and continue communications with the device.

Data Streams

The following sections describe the format of data streams that are sent and received by your AS/400 application programs when you use UDDS.

INIT-SELF Request: The format for the INIT-SELF request is:

```
X'0002020373F4'.

Hexadecimal data value      0002+0203+73+F4
                            ----+----+--+--
Byte position                1 2  3 4  5  6
```

Bytes 1 and 2 (X'0002')
Length of data passed from the AS/400 application to the finance controller application beginning in byte position 6 of the UDDS.

The minimum length of any UDDS is 2 bytes. Setting byte position 6 to X'F4' causes the extra byte at the end of the data stream to be ignored.

Bytes 3 and 4 (X'0203')
Length of data received from the finance application program. This length is set to X'0203' (decimal 515). It accepts a maximum of 512 bytes of data plus the three control characters before it.

Byte 5 (X'73')
UDDS control flag. It is set to a value representing a *write/read* operation.

Byte 6 (X'F4')
4700 control character. The 4-character value is an INIT-SELF request from the finance controller application program.

INIT-SELF Data Stream: The INIT-SELF data stream that the AS/400 application program receives from the finance controller application program is:

```
X'0101F301068100mmmm...mmmmF308xxxx...
  xxxx0000nnuuuu...uuuu'.
```

Bytes 1 and 2 (X'0101')
Reserved bytes.

Byte 3 (X'F3')
Formatted data will be present in the data stream.

Bytes 4 through 6 (X'010681')
SNA command for INIT-SELF request.

Byte 7 (X'00')
Reserved/control information.

Bytes 8 through 15 (represented: mmmm...mmmm)
Mode.

Bytes 16 through 25
Name of the destination logic unit (DLU).

Byte 16 (X'F3')
Type of logical unit.

Byte 17 (X'08')
Length of the symbolic name.

Bytes 18 through 25 (represented: xxxx...xxxx)
Symbolic name as either c'DTNCHXVS' for the 3694 processor, or c'SFSƀƀƀƀ' for the 4701 controller.

Byte 26 (X'00')
Requester ID length, no requester ID.

Byte 27 (X'00')
Password length, no password.

Bytes 28 through 50
User field.

Byte 28 (represented: nn)
Length of the user data.

Bytes 29 through 50 (represented: uuuu...uuuu)
User data.

The user field could contain the sign-on to the finance device. Your AS/400 application program should verify that the user ID is valid when the SBMFNCJOB command is not used. To do this, your AS/400 application can use a table of valid user IDs to approve the user ID passed in the user field of the INIT-SELF data stream. See “Work with User Table Command” on page 3-6 for information about user ID tables. (You may start the Display File Field Description (DSPFFD) command on the QUSRSYS/QFNUSRTBL file to determine the attributes of the user table file.) If the ID is not valid, the program may request the correct user ID.

The finance job assumes that the first 8 characters of the user data field contain the user ID from the controller application program.

The INIT-SELF request can pass 22 bytes of data from the controller application to the AS/400 application.

Positive Response to INIT-SELF Request:

The format for a positive response to INIT-SELF request is:

```
X'0002000071F1'.

Hexadecimal data value      0002+0000+71+F1
                            ----+----+--+--
Byte position                1 2  3 4  5  6
```

Bytes 1 and 2 (X'0002')

Length of data passed from the program to the finance controller application beginning in byte 6 of the UDDS.

The minimum length of any UDDS is set at a value of 2. Setting byte position 6 to X'F1' causes the extra byte at the end of the data stream to be ignored.

Bytes 3 and 4 (X'0000')

Length of data received from the finance application program. The length is set to 0 bytes to indicate that no data will be received from the controller.

Byte 5 (X'71')

UDDS control flag. It is set to a value representing a *write only* operation.

Byte 6 (X'F1')

4700 control character. The character 1 value indicates a positive response to the data received from the finance application program.

Negative Response to INIT-SELF Request:

The format for the negative response to INIT-SELF request is:

```
X'0008000071F208xx0000010681'.

Hexadecimal data value      0008+0000+71+F2+08xx0000+010681
                            ----+----+--+--+--------+------
Byte position                1 2  3 4  5  6  7    10 11  13
```

Bytes 1 and 2 (X'0008')

Length of data passed from the program to the finance controller application beginning in byte 6 of the UDDS.

Bytes 3 and 4 (X'0000')

Length of data received from the finance application program. This length is set to 0 bytes to indicate that no data will be received from the controller.

Byte 5 (X'71')

UDDS control flag. It is set to a value representing a *write only* operation.

Byte 6 (X'F2')

4700 control character. The character 2 value indicates a negative response to the data received from the finance controller application program.

Bytes 7 through 10 (X'08xx0000')

Returned to the controller as a negative response. The xx response code can be replaced with values, such as X'0F'—Not authorized to session or,X'35'—Invalid parameter (invalid data length).

For more information about the negative-response and the response code byte, see the *Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic*.

Bytes 11 through 13 (X'010681')

Returned to the controller application with bytes 7 through 10. This indicates that the negative response refers to an INIT-SELF data stream format.

TERM-SELF Data Stream:

The TERM-SELF data stream that the AS/400 application program receives from the controller application program is:

```
X'0101F301068300F308xxxx...xxxx'.
```

Bytes 1 and 2 (X'0101')

Reserved bytes.

Byte 3 (X'F3')

Formatted data follows in the data stream.

Bytes 4 through 6 (X'010683')
SNA command for TERM-SELF.

Byte 7 (X'00')
Reserved/control information.

Bytes 8 through 17
Name of the destination logical unit (DLU).

Byte 8 (X'F3')
Type of logical unit.

Byte 9 (X'08')
Length of the symbolic name.

Bytes 10 through 17 (represented: xxxx....xxxx)
Symbolic name either as c'DTNCHXVS' for the 3694 controller, or c'SFSƀƀƀƀ' for the 4701 controller.

Positive Response to TERM-SELF Request:

The format for the positive response to TERM-SELF request is:

```
X'0002000071F1'.

Hexadecimal data value      0002+0000+71+F1
                            ----+----+--+--
Byte position                1 2  3 4  5  6
```

Bytes 1 and 2 (X'0002')
Length of data passed from the program to the finance controller application beginning in byte 6 of the UDDS.

The minimum length of any UDDS is 2 bytes. Setting byte position 6 to X'F1' causes the extra byte at the end of the data stream to be ignored.

Bytes 3 and 4 (X'0000')
Length of data received from the finance controller application program. This length is set to 0 bytes to indicate that no data will be received from the controller.

Byte 5 (X'71')
UDDS control flag. It is set to a value representing a *write only* operation.

Byte 6 (X'F1')
4700 control character. The character 1 value indicates a positive response to the data received from the finance controller application program.

Negative Response to TERM-SELF Request:

The format for the negative response to TERM-SELF request is:

```
X'0008000071F208xx0000010681'.

Hexadecimal data value    0008+0000+71+F2+08xx0000+010681
                          ----+----+--+--+--------+------
Byte position              1 2  3 4  5  6  7    10  11 13
```

Bytes 1 and 2 (X'0008')
Length of data passed from the program to the finance controller application beginning in byte 6 of the UDDS.

Bytes 3 and 4 (X'0000')
Length of data received from the finance controller application program. This length is set to 0 bytes to indicate that no data will be received from the controller.

Byte 5 (X'71')
UDDS control flag. It is set to a value representing a *write only* operation.

Byte 6 (X'F2')
4700 control character. The character 2 value indicates a negative response to the data received from the finance controller application program.

Bytes 7 through 10 (X'08xx0000')
Returned to the controller as a negative response. The xx response code can be replaced with a value, such as X'16'–Session does not exist.

For more information about negative response code bytes, see the *Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic*.

Bytes 11 through 13 (X'010683')
Returned to the controller application with bytes 7 through 10 to indicate that the negative response refers to an INIT-SELF data stream format.

Logical Unit Status Command:

The Logical Unit Status (LUSTAT) command can be used by your program to report failures in the finance controller application. The format for the LUSTAT data stream is:

```
X'0006020373F3040000uuuu'.

Hexadecimal data value      0006+0203+73+F3+04+0000+uuuu
                            ----+----+--+--+--+----+----
Byte position                1 2  3 4  5  6  7  8 9 1011
```

Bytes 1 and 2 (X'0006')
Length of data passed from the program to the finance controller application beginning in byte 6 of the UDDS.

Bytes 3 and 4 (X'0203')
Length of data received from the finance controller application program. This length is set to 515 bytes to indicate that a maximum of 512 bytes of data and 3 bytes of control information is received from the controller.

Byte 5 (X'73')
UDDS control flag. It is set to a value representing a *write/read* operation.

Byte 6 (X'F3')
4700 control character. The character 3 value indicates that an LU status message is being sent to the finance controller application program.

Bytes 7 (X'04')
LUSTAT request code.

Bytes 8 and 9 (X'0000')
Status value for the LUSTAT data stream. These bytes are set to zero to indicate that the user status follows.

Bytes 10 and 11 (X'uuuu')
Status extension field. This file defines the exact message being sent to the finance controller. The values for this field include:

X'0000'	Session does not exist
X'0001'	Program ended normally
X'0002'	Program ended abnormally
X'0003'	Resource now available
X'0004'	Disable pending
X'10nn'	Procedure start failed, where nn indicates one of the following:

00	No additional information
01	Sign-on has invalid library name
02	Disk I/O error in security record
03	Job's starting ended by system operator
04	Resources not available to start job
05	Resource security file not found
06	Cannot log security information to history file
07	No user list in resource security file for library
08	Unauthorized request for user library
09	Invalid procedure name

A finance job started with the SBMFNCJOB command uses LUSTAT data streams with the user status fields of X'0002' and X'1009'. The X'0002' message is sent when a request for an application program by the program ID (included with the transaction) fails. The X'1009' message is sent when the program ID included with the transaction does not exist in the program table associated with the finance job.

For more information concerning the LUSTAT command and the status extension fields, see the *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic*.

3694 Communications with User-Defined Data Stream

Communication between the AS/400 application program and the CHX/3694 program is controlled by function-management-headers. For more information about function-management-headers, see the *Check Processing Executive/VS: Program Logic Manual*. For more information about programming for the 3694 controller, see the *Check Processing Executive/3694: Program Reference and Operations Manual* and the *Check Processing Executive/VS: Program Reference and Operations Manual*.

Function-Management-Headers: A 3694 function-management-header is a special record (or portion of a record) that contains control information for the data that follows. The first byte is the length of the header. The length is in hexadecimal values and includes the length byte. The header portion immediately follows the length byte.

The 3694 processor verifies the data before the function-management-header is sent to the AS/400 system. When the AS/400 system receives the data, it identifies the header and sets the data type byte to C'3' to indicate to the program that the header was received.

Input Data Format: The data (read by the program) has the following format when a function-management-header is received from the 3694 processor:

Bytes 1-2
Reserved

Byte 3
'3'

Byte 4
Function-management-header length (X'02')

Byte 5
Function-management-header identifier (X'80')

Bytes 6-7
Function-management-header type

Bytes 8-n
Application data

The data (read by the program) has the following format when a function-management-header is *not* received from the 3694 processor:

Bytes 1-2
Reserved

Byte 3
'1'

Bytes 4-n
Application data

Output Data Format: The data written by your AS/400 application program must have this format if the data contains a function-management-header:

Bytes 1-5
UDDS control information (see "Using User-Defined Data Stream" on page 6-14)

Byte 6
'6' (4700 control character)

Byte 7 (X'02')
Function-management-header length

Byte 8
Function-management-header identifier (X'80')

Bytes 9-10
Function-management-header type (defined in *CHX/VS Logic Manual*)

Bytes 11-n
Application data (defined in the *CHX/3694 Logic Manual*)

The data written by your application program must have this format if the data does *not* contain a function-management-header.

Bytes 1-5
UDDS control information (see "Using User-Defined Data Stream" on page 6-14)

Byte 6
'0' (4700 control character)

Bytes 7-n
Application data (defined in the *CHX/3694 Logic Manual*)

To prepare to read the data from the 3694 processor without writing any data, your application program should do a write/read operation with the following data stream:

Bytes 1-5
UDDS control information (see "Using User-Defined Data Stream" on page 6-14)

Bytes 1-2 (X'0002')
Length of data being passed to the 3694 application is set to a minimum value of 2 bytes

Bytes 3-4 (X'0200')
Length of data being received from the 3694 application is set to the maximum data length permitted

Byte 5 (X'73')
UDDS control flag is set to a value representing a *write/read* operation

Byte 6
'5' (4700 control character) invite device without writing

Note: Data is not actually sent to the 3694 processor. The data is only invited from that device.

Example of User-Defined Data Stream

Figure 6-17 shows a typical communications plan using UDDS.

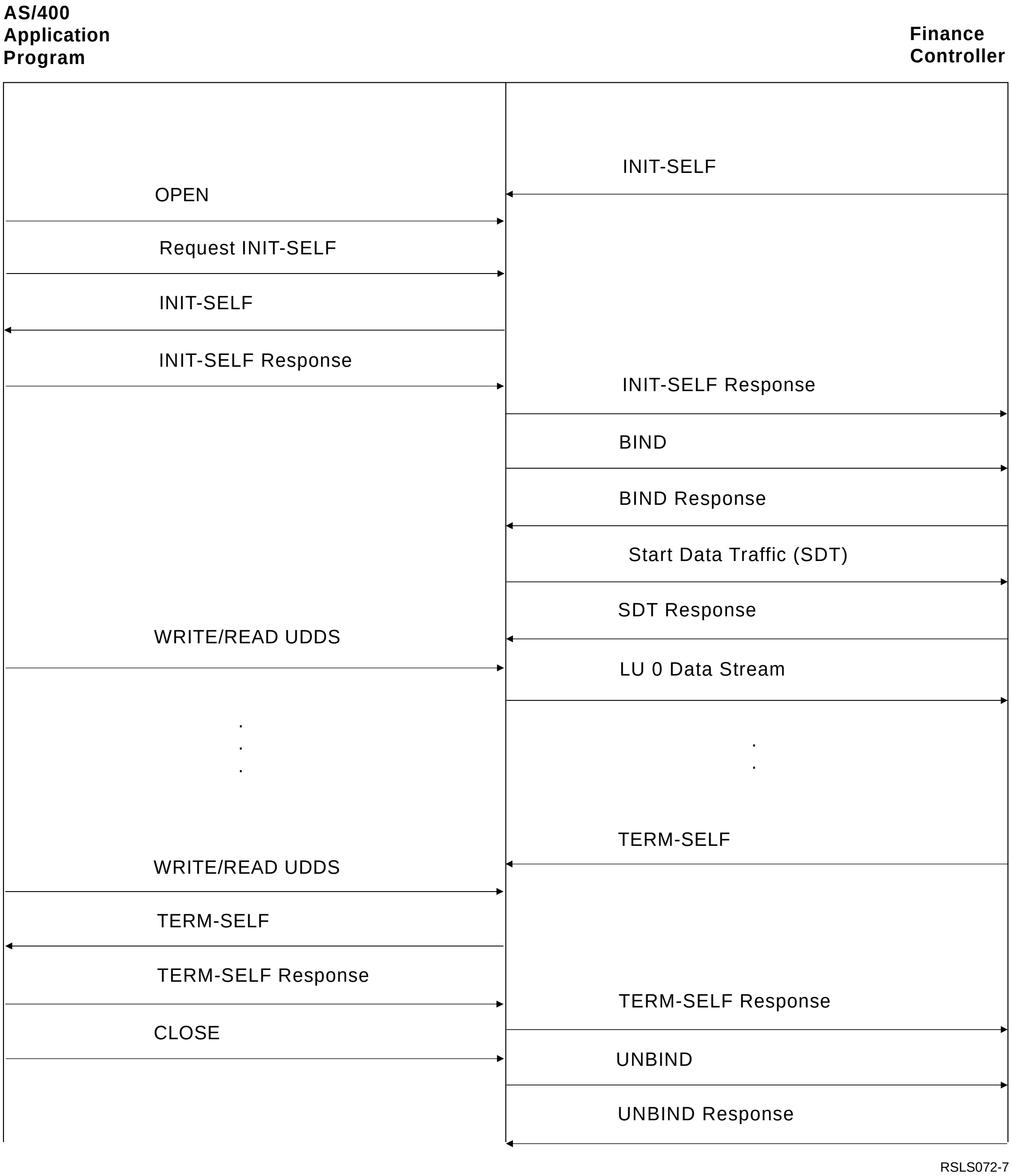


Figure 6-17. UDDS Communications Scenario

Chapter 7. Finance Considerations

This chapter describes considerations for programming, controllers, performance, and Systems Network Architecture (SNA) for finance communications.

Intersystem Communications Function

This section discusses considerations when using ICF finance.

Programming Applications

When planning your programs for use with finance communications, you must consider the protocol being used, the error recovery objectives, and the performance objectives.

ICF finance uses a **half-duplex** contention protocol. Half-duplex means that information can be sent in one direction at a time over the data link. Only when the information transmission ends can information be sent in the opposite direction. **Contention** means that you cannot control which program should send and which program should receive. If your AS/400 application program and the remote controller program both attempt to send data at the same time, the AS/400 program is the contention loser. Then the AS/400 program receives the data from the remote controller program and sends the data when the controller program finishes sending data.

Consider the following when using programs for ICF finance communications:

- Ensure your program checks the major and minor codes after every operation to determine whether the operation was a success or a failure. For more information about the error codes, see Appendix B, “Return Codes, Messages, and Sense Codes.”
- Use the information in the input/output (I/O) feedback area for your program. For more information about the I/O feedback area, see “Using Input/Output Feedback Area” on page 5-10.
- Remember, if the AS/400 program is the target program, it cannot start error recovery. If a permanent error occurs, the target program should finish any needed processing and end. The controller program is responsible for establishing the session again.

Performance

If you experience performance problems, performance improvements could occur when additional storage is moved from the base pool to the machine pool. For additional information concerning ways to improve your system performance, see the book, *Work Management*.

Buffering: Finance communications support buffers data sent by your program. The first record of a group is always sent immediately. The remaining records in a group are not sent until the communications buffer becomes full, a force-data function is specified, or the last record in a group is sent. Only single record groups are sent to a 3694 processor. Data sent on a write operation always results in a single record group being sent to the 3694 processor, regardless of the functions specified on the write operation.

Responses: A group of records that was partially sent can be completed by explicitly specifying the end-of-group (ENDGRP) function. A group of records also can be implicitly completed by a read, write-with-read, or a write-with-invite function.

For a 47xx or FBSS controller, if a group of records ended explicitly with the end-of-group function, a response is required from the remote controller program. The write operation does not end until the response is received. This may not be desirable when performance is a critical consideration. A response is not required from a 3694 processor, regardless of the functions specified on the write operation.

When a group of records is closed implicitly by a read operation, a write-with-invite function, or a write-with-read operation, a response is not required for the group from the remote controller.

When a group of records is sent on the system monitor session for a 4701 or 4702 controller, a response is always required from the controller before the write operation ends.

Prestart Jobs Using Program Start Requests: To minimize the time required to start a job when a program start request is received, you can use a prestart job entry to start a job on the AS/400 system before the controller program sends a program start request.

To use prestart jobs and to ensure programming changes are made in the prestart job program, you must define both communications and prestart job entries in the subsystem description. For more information about the prestart job entries, see the book, *ICF Programming*.

Program Initialization Parameters: If a program start request is received, each parameter received must be equal in length to the corresponding parameter specified in the AS/400 program. If the received parameter length exceeds the parameter length in the AS/400 program, truncation occurs. If the received parameter length is less than the parameter length in the AS/400 program, results that cannot be predicted could occur.

Security: If the data supplied on the INIT-SELF command fails security checking for any reason, the session will not be established. The INIT-SELF command may also fail due to a previous session not being completely reset, or due to parameters that are not valid on the INIT-SELF request that is received. The INIT-SELF command will be rejected.

A message describing the error and the sense code is issued to the QSYSOPR message queue.

For more information about sense codes, see Appendix B, “Return Codes, Messages, and Sense Codes.”

Non-Intersystem Communications Function

This section discusses considerations when using non-ICF finance.

Programming Applications

The following programming considerations apply to the non-ICF finance communications functions using finance input/output manager (FIOM), Submit Finance Job (SBMFNCJOB) command, and user-defined data stream (UDDS).

Finance Input/Output Manager (FIOM): When designing AS/400 application programs, you can use FIOM to do the following:

- To perform more than one write operation to the controller (rather than using the SBMFNCJOB command).
- To communicate interactively with the controller.
- To use a call interface (no communications operations).
- To write your own router program to handle data in a nonserial manner.

If you decide to use FIOM, you must override the QDFNDATA file by using the Override with Display File (OVRDSPF) command when defining devices to be used by the display file. Your application program must:

- Use the QDFNDATA file. If you do not, the FIOM routine receives a file-not-opened exception with CPD8289 (input/output exception received) and CPF8390 (errors occurred when running the program) messages. These messages are sent to your application program.
- Acquire and release finance devices used by the job, unless FIOM is being used with the SBMFNCJOB command.
- Open the QDFNDATA file as a shared file before a FIOM routine is started. The file should be closed by your application before returning control to the program that started your application. The SBMFNCJOB command opens and closes the file and acquires and releases the finance devices.

- Ensure the data length parameter passed to the QFNWRT, QFNWRTI, QFNREAD, and QFNREADI routines is initialized to a valid packed decimal number (15 5) to avoid receiving a CPD8286 (invalid format for data length parameter) error message.
- Ensure the data parameter is large enough to handle the maximum length of data that could be received from the finance controller application. A maximum of 512 bytes is allowed. See the XPF programmer's guide and the controller application guide because there may be requirements of fewer than 512 bytes. If the data parameter is not long enough, adjacent data space could be overwritten with financial data and give unexpected results.
- Process the send and receive data according to the format defined by the controller application and to ensure that the application program conforms to SNA communications rules.
- Handle any error recovery because errors received by the FIOM routines result in error messages sent to your application program.

Submit Finance Job (SBMFNCJOB) Command:

This command starts a continual BATCH job. The transaction programs receive controller data from and give data to the SBMFNCJOB command and send the data back to the controller. The SBMFNCJOB command handles incoming data from all devices serially. Only one transaction program can be running at one time.

Consider the following information about the SBMFNCJOB command when designing application programs:

- Avoid sending data directly from the application program to the controller application to help minimize finance job wait time. Let the finance job return data to the 4701 or 4702 application whenever possible.
- Start the QFNROUTE program directly, to provide interactive debugging of the application programs. The format of the command is:

  ```
  CALL PGM(QSYS/QFNROUTE) PARM(device-table-name
  program-table-name user-table-name
  'message-queue-name')
  ```

If you use the SBMFNCJOB interface to communicate with a controller, you must consider the hardware configuration of your AS/400 system. You must decide what devices to use, the number of devices per controller, and how these displays communicate.

To help you with these decisions, consider the following:

- Decide which finance controller application program to use by determining the amount of work that can be unloaded from the AS/400 system to the controller. The following factors affect this decision:
  - The amount of function supplied by the various applications at the controller level
  - The quantity and speed (due to the hardware configuration of your system) of transactions in which you require data to be passed to the AS/400 system
  - The amount of storage required for the application
  - Performance information supplied with each application
- Minimize unnecessary device acquires during the finance job starting phase by carefully changing the device tables. By balancing and distributing the processing load in the most efficient manner for the operating environment, the transaction processing you do at a later time improves. When designing the device tables, consider dividing devices in one of the following ways:
- By common functions, for example, placing all tellers on one job and all loan officers on another job.
- By controller, for example, placing all the devices on one controller assigned to one finance job.

Remember a finance job must acquire the first device it specifies in a device table to successfully start the finance job. The finance device must be active and not be in use by another job.

If a device is included in more than one device table, and the SBMFNCJOB interface is used, only one finance job can acquire that device.

Consider the possibility for I/O failure in your environment. If an unrecoverable I/O error occurs on a finance controller or line, the job started by the SBMFNCJOB command ends. This also ends communication with all devices associated with that job. For more information about I/O error handling capabilities of the SBMFNCJOB interface, see “Input/Output” on page 6-3.

Consider submitting multiple finance jobs. The jobs submitted handle transaction requests serially. The SBMFNCJOB command calls your transaction processing application and waits for control to return before it can process another request from any device associated with the finance job. Therefore, submitting more than one finance job reduces jobs waiting in a queue because of serial processing within one job.

As more devices are added to a device table, the program access group for the finance job using that device table increases, primarily due to the increased number of I/O buffers associated with the job. Therefore, submitting more than one finance job also reduces the group size of a single job accessing the program.

- Change the QFNC subsystem, job queue, and class to suit the needs of your particular operating environment. For example, you can change the QFNC class running priority to balance the workload of your system.
- Use the JOBD parameter of the SBMFNCJOB command to specify a job description having routing data other than QFNC specified on the QFNC job description. This allows you to specify different classes, and thus different running priorities and time slices, for individual finance jobs. The QFNC class is EXCPTY (20) and TIME SLICE (2000).
- Change this wait time by using the Create Class (CRTCLS) control language (CL) command to create a class with the DFTWAIT parameter set to the wait time you want. Display file QDFNDATA has been created with the WAITFILE parameter of (*CLS); therefore, the maximum amount of time spent trying to acquire a finance device is determined by the class associated with the finance job. The QFNC class specifies a default wait time of 30 seconds. If you reduce this wait time, the finance job may not have the ability to acquire a device in environments in which many devices are acquired or released at the same time.

  With high use of the system, performance improvements can occur when additional storage is moved from the base pool to the machine pool. For additional information concerning system adjustment, see the book, *Work Management*.

Finance support attempts error recovery if a finance job receives an I/O exception response. When an I/O exception response signals a finance job, the major/minor return code is retrieved from the message to determine the possibility of error recovery. Recoverable errors alert the finance job to try a recovery procedure. If the procedure is successful, processing continues normally. If the process is unsuccessful, the next action depends on the nature of the error.

Device errors result in the release of the affected device but other devices associated with the job continue processing. However, if recovery is not successful, controller or line errors end the job.

If you use the SBMFNCJOB command as the communications method between the AS/400 system and the controllers, special security exists, as discussed in the following considerations.

Granting Authority to Finance Objects: To keep financial information secure, the objects shipped with the system have restricted accessibility. Therefore, the following tasks must be completed before using the SBMFNCJOB interface.

- Use the Grant Object Authority (GRTOBJAUT) command to grant authority to the following users.

    Individual or group authorization to programmers who update the tables using the WRKDEVTBL, WRKPGMTBL, and WRKUSRTBL commands:

    ```
    GRTOBJAUT  OBJ(QSYS/WRKDEVTBL)  OBJTYPE(*CMD)
               USER(user-name)  AUT(*CHANGE)
    GRTOBJAUT  OBJ(QSYS/WRKPGMTBL)  OBJTYPE(*CMD)
               USER(user-name)  AUT(*CHANGE)
    GRTOBJAUT  OBJ(QSYS/WRKUSRTBL)  OBJTYPE(*CMD)
               USER(user-name)  AUT(*CHANGE)
    ```

    Individual or group authorization to operators who submit finance jobs and must be authorized to the SBMFNCJOB command:

    ```
    GRTOBJAUT  OBJ(QSYS/SBMFNCJOB)  OBJTYPE(*CMD)
               USER(user-name)  AUT(*CHANGE)
    ```

- Authorize the user profile QFNC access to any devices, programs, libraries, and files used by the finance jobs submitted with the SBMFNCJOB command. Specifying AUT(*CHANGE) is enough authority for these programs. An alternative to granting the required authority is to specify the QFNC profile as the owner of the objects.

- Determine if the finance support user ID sent in the data stream with the INIT-SELF is to be approved. If the user ID is approved, determine which user ID is allowed in each finance job. Use one user table for each finance job, one table for all finance jobs, or a combination of uniquely shared user tables. Define your finance user tables using the WRKUSRTBL command.

    For more information about the WRKUSRTBL command, see "Work with User Table Command" on page 3-6. Ensure that these programs do the following:

    - Develop your AS/400 transaction application programs.
    - Accept and return the parameters for the SBMFNCJOB command.
    - Accept and return data according to specifications defined by the finance controller application program.

    Once you develop your programs, describe which programs are to be used to process transactions by using the WRKPGMTBL command. Use one table for each finance job, one for all finance jobs, or a combination of unique and shared program tables. For more information about the WRKPGMTBL command, see "Work with Program Table Command" on page 3-7.

- Follow the security instructions described in "Security" on page 7-6.

Additional Security Considerations: To improve the security of your finance system, use the following guidelines:

- Submit jobs through the SBMFNCJOB command using the QFNC user profile. Similarly, QFNC owns the commands used for working with device, user, and program tables. *The password of the QFNC user profile should remain secure.*

- To avoid the possibility of external use, do not create a job description that does not refer to QFNC support.

- Use the Display Job Description (DSPJOBD) command to display the default job description for the SBMFNCJOB command. You can change attributes of this job description (job logging level) with the Change Job Description (CHGJOBD) command. Creating different job descriptions also restricts access to individual finance jobs.

- Use the work with table commands to restrict access to transaction processing programs and devices. For example, in Figure 7-1 on page 7-6, only Jones has access to Program Table, PROG X.

- Consider which library list that the finance job should use. The current library list of the user's job that performed the SBMFNCJOB command becomes the library list used for that finance job.

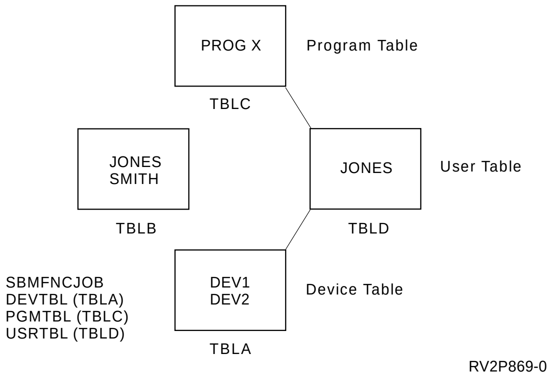


Figure 7-1. Accessing an Application Program

User-Defined Data Stream (UDDS): A **user-defined data stream (UDDS)** is a data stream in which the user has defined and embedded all device control characters. Consider the following information about UDDS when designing application programs:

- Ensure that your program correctly formats the data stream. If it does not, results that cannot be predicted could occur.
- Ensure that you compile your program with an externally described file that does not contain the USRDFN keyword. Then run the program with an externally described display containing the USRDFN keyword.

The MSGQ parameter (on the SBMFNCJOB command) expected by QFNROUTE is a 20-character variable (a 10-character message queue name immediately followed by a 10-character library name). The parameter represents the qualified name of the message queue to which finance messages are to be sent.

Security

After you have created your device, user, and program tables, have the person responsible for security on your system save backup versions of the QFNDEVTBL, QFNUSRTBL, and QFNPGMTBL files in another library.

Have the person responsible for security add the names of the libraries containing your AS/400 transaction application programs, and the names of any additional libraries required by those programs, to the original library list of QFNC. (QFNC is the job description found in the QGPL library under which the finance jobs run.)

Have the person responsible for security grant authorities required by user profile QFNC for programs used by your finance job. QFNC must be authorized for the device descriptions so that the devices can be acquired by the finance jobs, along with your AS/400 application programs and the libraries in which they reside.

Controller Applications

The following must be considered when you write your own application programs for use on the 4701, 4702, or FBSS controller.

To communicate successfully with finance communications, your controller application program must do the following:

- Handle and respond to the following commands received from the AS/400 system:
  - Activate Physical Unit (ACTPU)
  - Activate Logical Unit (ACTLU)
  - Bind (BIND)
  - Request for Start Data Traffic (SDT)
  - Clear (CLEAR)
  - Unbind (UNBIND)
- Start a session by sending the INIT-SELF request to the AS/400 system.
- End a session by sending the TERM-SELF request to the AS/400 system.
- Do not use brackets or the Change-of-Direction (CD) command during a session.
- Do not require a Bid (BID) sequence at the beginning of the session.

The AS/400 system requires that any device configured as a 3278 or a 3279 must accept extended data streams. Be sure that the emulation program running on the 4701, 4702, or FBSS controller has extended data stream support. If it does not have extended data stream support, configure for a 3277 device.

The device type you specify in the device description determines the form of data stream sent to the finance controller. The data stream can be one of the following:

- LU0 for device type 3624, 3694, 4704, or *FNCICF
- LU1 for device type 3287
- LU2 for device type 3277, 3278, or 3279

LU0 data streams allow printer data to be included with display data. LU1 data streams are sent to printers. The LU2 data streams require you to create 3270-type application displays. Be aware that translation and emulation could make the LU2 approach run slower than the LU0 support.

A 4701 or 4702 controller can do 5250 emulation (with a 4701- or 4702-based, 5250 emulation package), or can allow attachment of a personal computer emulating a finance device. Contact your IBM remarketer for information about additional finance configuration options.

Systems Network Architecture (SNA)

This section discusses the **Systems Network Architecture (SNA)** considerations for ICF and non-ICF finance communications. SNA describes the layered logical structure, formats, protocols, and operational sequences used for transmitting information units through networks, as well as controlling the configuration and operation of networks. The following data formats are used by SNA for the INIT-SELF command.

INIT-SELF Command Field Format

The INIT-SELF request starts an SNA session. Figure 7-2 shows the format needed for the INIT-SELF command.

Figure 7-2. Initiate-Self Request Field Format

Offset in Decimal	Description
0 through 2	Network service header: must be hex 010681.
3 through 12	Initiate header: must be hex 004040404040404040F3.
13	Length of destination logical unit name: must be hex 08
14 through 21	For 47xx and FBSS controllers, this is the primary logical unit name that is placed in the SNA BIND command. For the 3694 processor, this is the name of the AS/400 program to be started. It must be 8 bytes in length. If the program name is less than 8 bytes, the field must be padded with blanks.
22 through 23	Must be hex 0000.
24	Length (binary) of user data following this byte.
25 through n	User data field.

3694 Document Processor with ICF:

The 3694 processor follows the same procedure for initiation as for the other finance controllers, except the 3694 processor also sends the program name to be started on the INIT-SELF along with the security data. Then the program can communicate with the controller using the finance support.

On a secure AS/400 system, the INIT-SELF request that is received must include a valid user ID and password in the user data field of the INIT-SELF command. A program name is required in the destination logical unit (LU) field of the INIT-SELF.

If any data is included in the user data field of the INIT-SELF request, the field must be 10 bytes in length and formatted as shown in Figure 7-3 on page 7-8.

Figure 7-3. Format for User Data for 3694 INIT-SELF

Field	Description
1	Request code.
2 through 4	User ID; these 3 characters are added to USER to form the user identifier.
5 through 8	Password.
9 through 10	These 2 characters are added to LIB to form the library name. If these characters are 00, the current library list of the subsystem containing the communications entry is used.

47xx and FBSS Finance Controllers with Intersystem Communications Function:

The INIT-SELF user data field is also used by 47xx and FBSS controllers. On a secure system, the received INIT-SELF request must contain a user ID and password and can also contain a library name. Figure 7-4 shows the format of the user data field.

Figure 7-4. Format for User Data for 4701 and 4702 INIT-SELF

Field	Description
User_id, password[, library_name]	The maximum length of the user ID, password, and library name in the INIT-SELF request is 10 characters each. This is used to validate authority. The comma and library name following the password are optional.

Finance Controllers with Non-ICF Finance:

For information about non-ICF, see “Sending Data from the Finance Controller to the AS/400 System” on page 6-6.

If the SBMFNCJOB interface is used, the INIT-SELF must use the format in Figure 7-2 on page 7-7 with the following exceptions:

- Restrictions do not exist for the name of the destination logical unit.
- The library name is not present in the user data field.

If the SBMFNCJOB interface is not used, the format is determined by the user program.

BIND Command Field Format

The **BIND command** is used to start a session and to establish what protocol is followed for the current session. Figure 7-5 shows the BIND parameters for ICF finance to be used for each controller.

Figure 7-5. BIND Command Field Format for ICF Finance

Protocol	47xx and FBSS Controllers	4701 and 4702 System Monitor	3694 Document Processor
Function management profile	04	04	04
Transmission profile	04	04	04
Primary logical unit protocol	B0	A0	10
Secondary logical unit protocol	B0	B0	30
Common protocol	4040	4040	4040

Figure 7-6 on page 7-9 shows the BIND parameters for non-ICF finance to be used for each controller.

Figure 7-6. BIND Command Field Format for Non-ICF Finance

Protocol	4701 and 4702 Controllers	3694 Document Processor
Function management profile	04	04
Transmission profile	04	04
Primary logical unit protocol	10	10
Secondary logical unit protocol	B0	30
Common protocol	4000	4000

Appendix A. Language Operations, DDS Keywords, and System-Supplied Formats

This appendix provides information about the following:

- Valid communications operations supported by ICF finance
- Valid finance communications operations supported and the associated high-level language operations
- Data description specifications (DDS) processing keywords
- System-supplied formats

Using Language Operations

You can use ICF operations and high-level program languages to use finance communications. This discussion defines the operations used for finance and the differences in the language statements for C Set ++ for OS/400, ILE C/400, COBOL/400, and RPG/400 programming languages.

Intersystem Communications Function Operations

Figure A-1 provides a brief description of the ICF operations supported by finance communications.

Figure A-1. ICF Operations Supported by Finance Communications

ICF Operation	Description
Open	Opens the ICF file.
Acquire	Establishes an ICF session between the application and the remote location.
Get-attributes	Determines the status of the session.
Read	Obtains data from a specific session.
Read-from-invited- program-devices	Obtains data from any session responding to an invite function.
Write	Passes data records from the local program to the remote program.
Write/Read	Allows a write operation followed by a read operation.
Release	Attempts to end an ICF session.
Close	Closes the ICF file.

Intersystem Communications Function Language Statements

Figure A-2 provides a list of ICF operations supported by finance communications and the equivalent language statements needed to run these operations.

Figure A-2. ICF Operations and Equivalent Language Statements

ICF Operation	ILE RPG/400 Function	ILE COBOL/400 Procedure Statement	C Functions[2]
Open	OPEN	OPEN	fopen or_Ropen
Acquire	ACQ	ACQUIRE	_Racquire
Get-attributes	POST	ACCEPT	_Rdevatr
Read	READ	READ	fread or _Rreadn
Read-from- invited-program- devices	READ[1]	READ[1]	fread or _Rreadindv
Write	WRITE	WRITE	fwrite or _Rwrite
Write/read	EXFMT	Not supported	_Rwriterd
Release	REL	DROP	_Rrelease
Close	CLOSE	CLOSE	fclose or _Rclose

Notes:

[1] A read operation can be directed either to a specific program device or to any invited program device. The support provided by the compiler that you are using determines whether to issue an ICF read or read-from-invited-program-devices operation, based on the format of the read operation. For example, if a read operation is issued with a format or display specified, the read operation is interpreted as an ICF read operation. Refer to the appropriate language reference book for more information.

[2] *C Functions* represents functions for both ILE C/400 and C Set ++ for OS/400 languages. Also, both languages are case sensitive.

Data Description Specifications Keywords

Read and write operations use a record containing DDS keywords. These keywords allow you to use more specific communications functions with the read and write operations. Figure A-3 shows all the keywords supported by ICF finance communications.

Figure A-3. Valid DDS Keywords for ICF Finance Communications

DDS Keyword	Description
CANCEL[1]	Cancels a group of records that was partially sent.
CNLINVITE	Cancels any valid invite for which data has not yet been received.
ENDGRP	Indicates the end of a user-defined group of records.
EOS	Specifies an end-of-session function.
FAIL	Sends a fail indication to the remote system.
FMH	Informs the remote program that a function-management-header is being sent.
FRCDTA	Sends data immediately for the write operation and also sends data currently in the communications buffer, without waiting for the buffer to become full.
INVITE	Schedules an invite request.
NEGRSP	Informs the remote system that the data received is not valid.
RCVENDGRP	Indicates that the end of a user-defined group (chain) of records was received.
RCVFMH	Indicates to the program that a function-management-header was received.
RCVNEGRSP	Indicates that the remote program sent a negative response.
RECID	Allows the data content to identify the record format to use for receiving data.
TIMER	Allows the user to specify an interval of time to wait before a read-from-invited-program-devices operation receives a timer-expired return code.
VARLEN	Specifies that the length of the user data is defined in the 5 bytes of the specified field.

[1] Not valid for a 3694 processor.

System-Supplied Formats

Figure A-4 shows the functions and operations performed by the system-supplied formats that are valid for finance communications.

Figure A-4. System-Supplied Formats

System- Supplied Format	Equivalent DDS Keyword	Description
$$CANL[1]	CANCEL, INVITE	Send SNA cancel, then invite
$$CANLNI[1]	CANCEL	Send SNA cancel
$$CNLINV	CNLINVITE	Cancel an invite
$$EOS	EOS	End of session
$$FAIL	FAIL	Fail
$$NRSP	NEGRSP, INVITE	Negative response, then invite
$$NRSPNI	NEGRSP	Negative response
$$SEND	INVITE	Write then invite, or invite
$$SENDE	ENDGRP	Write with end-of-group
$$SENDFM	FMH, Invite	Write with function-management-header, then invite
$$SENDNF	FMH	Write with function-management-header
$$SENDNI	No DDS keyword	Write
$$TIMER	TIMER	Set timer

[1] Not valid for a 3694 document processor.

Appendix B. Return Codes, Messages, and Sense Codes

Return Codes

This section describes all the return codes that are valid for finance communications. These return codes are set in the I/O feedback area of the ICF file; they report the results of each I/O operation issued by your application program. Your program should check the return code and act accordingly. Refer to your high-level language book for more information on how to access these return codes.

Each return code is a four-digit hexadecimal value. The first two digits contain the *major code*, and the last two digits contain the *minor code*.

With some return codes, a message is also sent to the job log or the system operator message queue (QSYSOPR). You can refer to the message for additional information.

Notes:

1. In the return code descriptions, *your program* refers to the local AS/400 application program that issues the operation and receives a return code from ICF communications. The *remote program* refers to the application program on the remote system with which your program is communicating through ICF.
2. Several references to input and output operations are made in the descriptions. These operations can include DDS keywords and system-supplied formats, which are listed in Appendix A.

Major Code 00

Major Code 00 – Operation completed successfully.

Description: The operation issued by your program completed successfully. Your program may have sent or received some data, or may have received a message from the remote system.

Action: Examine the minor return code and continue with the next operation.

Code **Description/Action**

0000 **Description:** For output operations issued by your program, 0000 indicates that the last output operation completed successfully and that your program can continue to send data.

Action: Issue an input or output request.

0001 **Description:** Your program has successfully invited the finance session.

Action: Issue a read-from-invited-program-devices operation.

0003 **Description:** On a successful input operation, your program received a group of records.

Action: Issue an input operation to receive the next group of records, or issue an output operation.

0007 **Description:** On a successful input operation, your program received a group of records with a function-management-header (FMH).

Action: Issue an input or output operation.

Major Code 02

Major Code 02 – Input operation completed successfully, but your job is being ended (controlled).

Description: The input operation issued by your program completed successfully. Your program may have received some data or a message from the remote system. However, your job is being ended (controlled).

Action: Your program should complete its processing and end as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

Code **Description/Action**

0200 **Description:** On a successful input operation, your program received some data. Also, your job is being ended (controlled).

Action: Your program can continue to receive data, or it can send data to the remote program. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

0203 **Description:** On a successful input operation, your program received a group of records. Also, your job is being ended (controlled).

Action: Your program can issue an input operation to receive the next group of records, or it can issue an output operation. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

0207 **Description:** On a successful input operation, your program received a group of records with a function-management-header (FMH). Also, your job is being ended (controlled).

Action: Your program can issue an input or output operation. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

Major Code 03

Major Code 03 – Input operation completed successfully, but no data received.

Description: The input operation issued by your program completed successfully, but no data was received.

Action: Examine the minor return code for a function-management-header or a timer indication, and continue with the next operation.

Code **Description/Action**

0303 **Description:** On a successful input operation, your program received an end-of-group indication without any data.

Action: Issue an input or output operation.

0309 **Description:** On a read-from-invited-program-devices operation, your program did not receive any data. Also, your job is being ended (controlled).

Action: Your program can continue processing. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

Messages:

```
CPF4741 (Notify)
```

0310 **Description:** On a read-from-invited-program-devices operation, the time interval specified by a timer function in your program or by the WAITRCD value specified for the ICF file expired.

Action: Issue the intended operation after the specified time interval has ended. For example, if you were using the time interval to control the length of time to wait for data, you can issue another read-from-invited-program-devices operation to receive the data.

Note: Since no specific program device name is associated with the completion of this operation, the program device name in the common I/O feedback area is set to *N. Therefore, your program should not make any checks based on the program device name after receiving the 0310 return code.

Messages:

```
CPF4742 (Status)
CPF4743 (Status)
```

Major Code 04

Major Code 04 – Output exception occurred.

Description: An output exception occurred because your program attempted to send data when it should be receiving data or a response indication. The data from your output operation was not sent. You can attempt to send the data later.

Action: Issue an input operation to receive the data or response indication.

Code **Description/Action**

0412 **Description:** An output exception occurred because your program attempted to send data or to cancel an invite function when it should be receiving data or a response indication that was sent by the remote program. The data from your output operation was not sent to the remote system. Your program can attempt to send the data later.

Action: Issue an input operation to receive the data or response indication.

Messages:

```
CPF4750 (Notify)
CPF5076 (Notify)
```

Major Codes 08 and 11

Major Codes 08 and 11 – Miscellaneous program errors occurred.

Description: The operation just attempted by your program was not successful. The operation may have failed because it was issued at the wrong time.

Action: Refer to the minor code description for the appropriate recovery action.

Code **Description/Action**

0800 **Description:** The acquire operation just attempted by your program was not successful. Your program tried to acquire a program device that was already acquired and is still active.

Action: If the session associated with the original acquire operation is the one needed, your program can begin communicating in that session since it is already available. If you want a different session, issue another acquire operation for the new session by specifying a different program device name in the PGMDEV parameter of the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command that precedes the program.

Messages:

```
CPD4077 (Diagnostic)
CPF5041 (Status)
CPF50A0 (Status)
```

1100 **Description:** The read-from-invited-program-devices operation just attempted by your program was not successful because your program tried this operation when no program devices were invited and no timer function was in effect.

Action: Issue an invite function (or a combined operation that includes an invite) followed by a read-from-invited-program-devices operation.

Messages:

```
CPF4740 (Notify)
```

Major Code 34

Major Code 34 – Input exception occurred.

Description: The input operation attempted by your program was not successful. The data received was too long for your program's input buffer or was not compatible with the record format specified on the input operation.

Action: Refer to the minor code description for the appropriate recovery action.

Code **Description/Action**

3401 **Description:** The input operation issued by your program was not successful because the length of the data record sent by the remote system was longer than the length specified for your program's input buffer. The length of the data record received from the remote system, if available, is in the actual-record-length field in the I/O feedback area.

Action: Issue another input operation if your program can specify a record size large enough to receive the data, plus any indicators for a file without a separate indicator area. Otherwise, you should close the file, end your program, correct the record size, then run your program again.

Messages:

```
CPF4768 (Notify)
```

3441 **Description:** A valid record format name was specified with format selection type *RECID. However, although the data received matched one of the record formats in the ICF file, it did not match the format specified on the read operation.

Action: Correct your program to issue a read operation that does not specify a record format name, or specify the correct record format name to process the data based on the format selection option for the file.

Messages:

```
CPF5058 (Notify)
```

3451 **Description:** Your program specified a file record size that was not large enough for the indicators to be included with the data sent by the remote program (for a file defined with a nonseparate indicator area). Your program did not receive any data. For a file using a nonseparate indicator area, the actual record length field in the device-dependent I/O feedback area contains the number of indicators specified by the record format.

Action: End the session; close the file; correct the file record size; then open the file again.

Messages:

```
CPF4768 (Notify)
```

Major Code 80

Major Code 80 – Permanent system or file error (irrecoverable).

Description: An irrecoverable file or system error has occurred. The underlying communications support may have ended and your session has ended. If the underlying communications support ended, it must be established again before communications can resume. Recovery from this error is unlikely until the problem causing the error is detected and corrected.

Action: You can perform the following general actions for all 80xx return codes. Specific actions are given in each minor code description.

- Close the file, open the file again, then establish the session. If the operation is still not successful, your program should end the session.
- Continue local processing.
- End.

Note: If the session is started again, it starts from the beginning, not at the point where the session error occurred.

Code **Description/Action**

8081 **Description:** The operation attempted by your program was not successful because a system error condition was detected.

Action: Your communications configurations may need to be varied off and then on again. Your program can do one of the following:

- Continue local processing.
- Close the ICF file, open the file again, and establish the session again.
- End.

Messages:

```
CPF4170 (Escape)
CPF4510 (Escape)
CPF5197 (Escape)
CPF5244 (Escape)
CPF5257 (Escape)
CPF5274 (Escape)
CPF5346 (Escape)
CPF5355 (Escape)
```

8082 **Description:** The operation attempted by your program was not successful because the device supporting communications between your program and the remote location is not usable. For example, this may have occurred because communications were stopped for the device by a Hold Communications Device (HLDCMNDEV) command. Your program should not issue any operations to the device.

Action: Communications with the remote program cannot resume until the device has been reset to a varied on state. If the device has been held, use the Release Communications Device (RLSCMNDEV) command to reset the device. If the device is in an error state, vary the device off and then on again. Your program can attempt to establish the session again, continue local processing, or end.

Messages:

```
CPF4744 (Escape)
CPF5269 (Escape)
```

80B3 **Description:** The open operation issued by your program was not successful because the ICF file is in use by another process.

Action: Wait for the file to become available, then issue another open operation. Otherwise, your program may continue processing, or it can end.

Consider increasing the WAITFILE parameter with the Change ICF File (CHGICFF) or Override ICF File (OVRICFF) command to allow more time for the file resources to become available.

Messages:

```
CPF4128 (Escape)
```

80EB **Description:** The open operation attempted by your program was not successful due to one of the following:

- Your program used an option of update or delete to open the file, but that option is not supported by the program device.
- Your program requested both blocked data and user buffers on an open option, but these formats cannot be selected together.
- Your program tried to open a source file, but the file was not created as a source file.
- There is a mismatch on the INDARA keyword between your program and the ICF file as to whether or not a separate indicator area should be used.
- The file was originally opened as a shared file; however, no program devices were ever acquired for the file before your program attempted the current open operation.

Action: After performing one of the following actions, your program can try the open operation again:

- If the update and delete options are not supported for the program device, use an option of input, or output, or both.
- If your program tried selecting user buffers and blocked data together, it should try selecting one or the other, but not both.
- If your program tried to open a non-source file as a source file, either change the file name or change the library name.

- If there was a mismatch on the INDARA keyword, either correct the file or correct your program so that the two match.
- If no program devices were previously acquired for a shared file, acquire one or more program devices for the file.

Messages:

```
CPF4133 (Escape)
CPF4156 (Escape)
CPF4238 (Escape)
CPF4250 (Escape)
CPF4345 (Escape)
CPF5522 (Escape)
CPF5549 (Escape)
```

80ED **Description:** The open operation attempted by your program was not successful because there is a record format level mismatch between your program and the ICF file.

Action: Close the file. Compile your program again to match the file level of the ICF file, or change or override the file to LVLCHK(*NO); then open the file again.

Messages:

```
CPF4131 (Escape)
CPF4564 (Escape)
```

80EF **Description:** Your program attempted an open operation on a file or library for which the user is not authorized.

Action: Close the file. Either change the file or library name on the open operation, or obtain authority for the file or library from your security officer. Then issue the open operation again.

Messages:

```
CPF4104 (Escape)
```

80F8 **Description:** The open operation attempted by your program was not successful because one of the following occurred:

- The file is already open.
- The file is marked in error on a previous return code.

Action:

- If the file is already open, close the file and end your program. Remove the duplicate open operation from your program, then issue the open operation again.
- If the file is marked in error, your program can check the job log to see what errors occurred previously, then take the appropriate recovery action for those errors.

Messages:

```
CPF4132 (Escape)
CPF5129 (Escape)
```

Major Code 81

Major Code 81 – Permanent session error (irrecoverable).

Description: An irrecoverable session error occurred during an I/O operation. Your session cannot continue and has ended. Before communications can resume, the session must be established again by using an acquire operation or another program start request. Recovery from this error is unlikely until the problem causing the error is detected and corrected. Operations directed to other sessions associated with the file should work.

Action: You can perform the following general actions for all 81xx return codes. Specific actions are given in each minor return code description.

If your program initiated the session, you can:

- Correct the problem and establish the session again. If the operation is still not successful, your program should end the session.
- Continue processing without the session.
- End.

If your session was initiated by a program start request from the remote program, you can:

- Continue processing without the session.
- End.

Several of the minor codes indicate that an error condition must be corrected by changing a value in the communications configuration or in the file.

- To change a parameter value in the communications configuration, vary the configuration off, make the change to the configuration description, then vary the configuration on.
- To change a parameter value in the file, use the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.

  Note: When a parameter can be specified both in the ADDICFDEVE or OVRICFDEVE command and in the configuration, the value in the ADDICFDEVE or OVRICFDEVE command overrides the value specified in the configuration (for your program only). Therefore, in some cases, you may choose to make a change with the ADDICFDEVE or OVRICFDEVE command rather than in the configuration.

Several other minor codes indicate a line or remote system error and may require an operator to correct the error.

Note: If the session is started again, it starts from the beginning, not at the point where the session error occurred.

Code **Description/Action**

8140 **Description:** A cancel reply was received from your program or from the operator in response to a notify message, or was the result of a system default, causing the session to be ended. The session is no longer active.

Action: If your program started the session, issue an acquire operation to start the session again. If your program was started by a program start request, it can continue local processing or end.

Messages:

```
CPF5104 (Escape)
```

8191 **Description:** A permanent line or controller error occurred on an input or output operation, and the system operator attempted recovery in response to the error message. You can learn what type of line error occurred by checking the system operator's message queue. The session has ended. Data may have been lost.

Action: If your program started the session, issue an acquire operation to start the session again. If your program was started by a program start request from the remote program, it can continue local processing or end.

Messages:

```
CPF4146 (Escape)
CPF5128 (Escape)
CPF5342 (Escape)
CPF5344 (Escape)
```

8197 **Description:** On an input or output operation, the remote system ended the transmission abnormally because it could not continue the session. The session has ended.

Action: If your program started the session, issue an acquire operation to start the session again. If your program was started by a program start request from the remote program, it can continue local processing or end.

Messages:

```
CPF5167 (Escape)
CPF5241 (Escape)
```

81A3 **Description:** The session ended abnormally because of an SNA request shutdown, request recovery, or UNBIND command from the remote controller.

Action: Determine the reason for the error in the remote controller program. Correct the error, then start the session again.

Messages:

```
CPF5167 (Escape)
```

81A4 **Description:** An SNA protocol violation occurred on the input or output operation attempted by your program. A negative-response with sense data was sent to the controller.

Action: Examine the sense data in the associated message to determine the protocol error. Correct the error, then try the operation again. For more information on sense data, see the *Systems Network Architecture Reference Summary* book.

Messages:

CPF5248 (Escape)

81AD **Description:** The input or output operation issued by your program was not successful because the SDLC frame size was not large enough to contain the RU size. Either this was a configuration error, or the frame size was changed to a smaller value by the Exchange ID (XID) command.

Action: End your program. The SDLC frame size is specified in the MAXFRAME parameter on the controller description, and the RU size is specified in the MAXLENRU parameter on the device description. Verify that these configuration parameters are correct and, if necessary, reduce the RU size or increase the frame size. If changes to the configuration must be made, first vary the device off, then on again. Try to run your program again.

Messages:

CPF5341 (Escape)

81BA **Description:** Your program received a data record whose length exceeds the maximum user record length.

Action: Verify that the remote program sent the correct data. If so, end your program. Increase the value of the maximum record length (MAXRCDLEN) parameter in the ICF file, and increase the size of the input buffer on the record format to be used for the input operation, then try running your program again.

Messages:

CPF5205 (Escape)

81E9 **Description:** An input operation was issued and the format selection option for the ICF file was *RECID, but the data received did not match any record formats in the file. There was no format in the file defined without a RECID keyword, so there was no default record format to use. The session has ended.

Action: Verify that the data sent by the remote program was correct. If the data was not correct, have the operator on the remote system change the remote program to send the correct data. If the data was correct, add a RECID keyword definition to the file that matches the data, or define a record format in the file without a RECID keyword so that a default record format can be used on input operations. If your program started the session, use another acquire operation to start the session again. If a program start request started your program, continue local processing or end.

Messages:

CPF5291 (Escape)

Major Code 82

Major Code 82 – Open or acquire operation failed.

Description: Your attempt to establish a session was not successful. The error may be recoverable or permanent, and recovery from it is unlikely until the problem causing the error is detected and corrected.

Action: You can perform the following general actions for all 82xx return codes. Specific actions are given in each minor code description.

If your program was attempting to start the session, you can:

- Correct the problem and attempt to establish the session again. The next operation could be successful only if the error occurred because of some temporary condition such as the communications line being in use at the time. If the operation is still not successful, your program should end.
- Continue processing without the session.
- End.

If your session was initiated by a program start request from the remote program, you can:

- Correct the problem and attempt to connect to the requesting program device again. If the operation is still not successful, your program should end.
- Continue processing without the session.
- End.

Several of the minor codes indicate that an error condition must be corrected by changing a value in the communications configuration or in the file.

- To change a parameter value in the communications configuration, vary the configuration off, make the change to the configuration description, then vary the configuration on.
- To change a parameter value in the file, use the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.

  Note: When a parameter can be specified both in the ADDICFDEVE or OVRICFDEVE command and in the configuration, the value in the ADDICFDEVE or OVRICFDEVE command overrides the value specified in the configuration (for your program only). Therefore, in some cases, you may choose to make a change with the ADDICFDEVE or OVRICFDEVE command rather than in the configuration.

If no changes are needed in your file or in the configuration (and depending on what the return code description says):

- If the attempted operation was an acquire, issue the acquire operation again.
- If the attempted operation was an open, close the file and issue the open operation again.

Code **Description/Action**

8209 **Description:** The open or acquire operation issued by your program was not successful because a prestart job is being canceled. One of the following may have occurred:

- An End Job (ENDJOB), End Prestart Job (ENDPJ), End Subsystem (ENDSBS), End System (ENDSYS), or Power Down System (PWRDWNSYS) command was being issued.
- The maximum number of prestart jobs (MAXJOBS parameter) was reduced by the Change Prestart Job Entry (CHGPJE) command.
- The value for the maximum number of program start requests allowed (specified in the MAXUSE parameter on the ADDPJE or CHGPJE command) was exceeded.
- Too many unused prestart jobs exist.
- The prestart job had an initialization error.

Action: Complete all processing and end your program as soon as possible. Correct the system error before starting this job again.

Messages:

```
CPF4292 (Escape)
CPF5313 (Escape)
```

8233 **Description:** A program device name that was not valid was detected. Either an ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command was not run, or the program device name in your program does not match the program device name specified in the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command for the session being acquired. The session was not started.

Action: If the error was in your program, change your program to specify the correct program device name. If an incorrect identifier was specified in the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, specify the correct value in the PGMDEV parameter.

Messages:

```
CPF4288 (Escape)
CPF5068 (Escape)
```

8281 **Description:** On an unsuccessful open or acquire operation, a system error condition was detected. For example, the file may previously have been in error, or the file could not be opened due to a system error.

Action: Your communications configurations may need to be varied off and then on again. Your program can do one of the following:

- Continue local processing.
- Close the ICF file, open the file again, and acquire the program device again. However, if this results in another 8281 return code, your program should close the file and end.
- Close the file and end.

Messages:

```
CPF4143 (Escape)
CPF4168 (Escape)
CPF4182 (Escape)
```

CPF4231 (Escape)
CPF4254 (Escape)
CPF4265 (Escape)
CPF4304 (Escape)
CPF4369 (Escape)
CPF4370 (Escape)
CPF4375 (Escape)
CPF5197 (Escape)
CPF5202 (Escape)
CPF5244 (Escape)
CPF5250 (Escape)
CPF5257 (Escape)
CPF5274 (Escape)
CPF5317 (Escape)
CPF5318 (Escape)
CPF5355 (Escape)

8282 **Description:** The open or acquire operation attempted by your program was not successful because the device supporting communications between your program and the remote location is not usable. For example, this may have occurred because communications were stopped for the device by a Hold Communications Device (HLDCMNDEV) command. Your program should not issue any operations to the device. The session was not started.

Action: Communications with the remote program cannot resume until the device has been reset to a varied on state. If the device has been held, use the Release Communications Device (RLSCMNDEV) command to reset the device. If the device is in an error state, vary the device off, then on again. Your program can attempt to acquire the program device again, continue local processing, or end.

Messages:

CPF4298 (Escape)
CPF5269 (Escape)

8291 **Description:** A permanent line or controller error occurred on an unsuccessful open or acquire operation, and the system operator took a recovery option in response to the error message. The session was not started.

Action: If your program was attempting to start the session, it can try the acquire operation again. If your program was started by a program start request from the remote program, your program can continue local processing or end.

Messages:

CPF4193 (Escape)
CPF4261 (Escape)
CPF5260 (Escape)
CPF5342 (Escape)
CPF5344 (Escape)

8297 **Description:** An SNA TERM-SELF or UNBIND request was received, while your program was attempting to establish a communications session with the remote controller.

Action: Try the open or acquire operation again, continue local processing, or end your program.

Messages:

```
CPF4178 (Escape)
CPF5167 (Escape)
CPF5241 (Escape)
```

82A2 **Description:** Your program issued an SNA INIT-SELF request for a finance remote location or device description that did not contain valid authorization data. One of the following occurred:

- The user ID or password was not supplied.
- The specified user ID was not found on the system.
- The specified password was not valid for this user ID.
- The specified user ID was not authorized to use this device description.

Action: Verify that your program specifies the correct user ID or password on the INIT-SELF command, or create a user profile with the correct user ID and password. If the user is not authorized to the device, use the Grant Object Authority (GRTOBJ) command to authorize the user.

Messages:

```
CPF4177 (Escape)
CPF5251 (Escape)
```

82A4 **Description:** An SNA protocol violation occurred on the open or acquire operation attempted by your program. A negative-response with sense data was sent to the controller.

Action: Examine the sense data in the associated message to determine the protocol error. Correct the error, then try the operation again. For more information on sense data, see the *Systems Network Architecture Reference Summary* book.

Messages:

```
CPF4141 (Escape)
CPF5248 (Escape)
```

82A6 **Description:** One of the following occurred:

- A negative-response with sense data was received when the Systems Network Architecture (SNA) BIND or Start Data Traffic (SDT) command was sent to the user to start the session.
- The BIND or SDT command did not end within the time limit specified using the wait time (WAITFILE) parameter on the CRTICFF, CHGICFF, or OVRICFF command.

The session was not started.

Action: Close the file. Examine the associated messages for SNA sense data received when the BIND or SDT command failed, and verify that the local and remote configurations are compatible. Determine why the remote controller did not respond within the time limit. If the time limit is too short, increase the value specified in the WAITFILE parameter on the CHGICFF or OVRICFF command. Correct the error and run the program again.

Messages:

```
CPF4142 (Escape)
CPF4254 (Escape)
CPF4333 (Escape)
CPF4527 (Escape)
CPF5202 (Escape)
CPF5240 (Escape)
CPF5538 (Escape)
```

82A7 **Description:** The open or acquire operation attempted by your program was not successful because the specified program device was already in use. The session was not started.

Action: Your program can wait for the program device to become available, then try the open or acquire operation again. Otherwise, it can continue local processing or end.

Messages:

```
CPF4106 (Escape)
CPF5507 (Escape)
```

82A8 **Description:** The acquire operation attempted by your program was not successful because the maximum number of program devices allowed for the ICF file has been reached. The session was not started.

Action: Your program can recover by releasing a different program device and issuing the acquire operation again. If more program devices are needed, close the file and increase the MAXPGMDEV value for the ICF file.

Messages:

```
CPF4745 (Diagnostic)
CPF5041 (Status)
```

82A9 **Description:** The acquire operation issued by your program to a *REQUESTER device was not successful due to one of the following causes:

- Your program has already acquired the *REQUESTER device.
- The *REQUESTER device was released because an end-of-session was requested.
- The job does not have a *REQUESTER device; that is, the job was not started by a program start request.
- A permanent error occurred on the session.

Action:

- If the *REQUESTER device is already acquired and your program expects to communicate with the *REQUESTER device, use the program device that acquired the *REQUESTER.
- If your program released its *REQUESTER device, correct the error that caused your program to release its *REQUESTER device before trying to acquire it.
- If this job does not have a *REQUESTER device, correct the error that caused your program to attempt to acquire a *REQUESTER device.

- If a permanent error caused the acquire operation to fail, verify that your program correctly handles the permanent error return codes (80xx, 81xx) it received on previously issued input and output operations. Because your program was started by a program start request, your program cannot attempt error recovery after receiving a permanent error return code. It is the responsibility of the remote program to initiate error recovery.

Messages:

```
CPF4366 (Escape)
CPF5380 (Escape)
CPF5381 (Escape)
```

82AA **Description:** The open or acquire operation attempted by your program was not successful because the remote location name specified on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command does not match any remote location configured on the system. The session was not started.

Action: Your program can continue local processing, or close the file and end. Verify that the name of the remote location is specified correctly in the RMTLOCNAME parameter on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.

Messages:

```
CPF4103 (Escape)
CPF4363 (Escape)
CPF4364 (Escape)
CPF4747 (Escape)
CPF5378 (Escape)
CPF5379 (Escape)
```

82AB **Description:** The open or acquire operation attempted by your program was not successful because the device description for the remote location was not varied on. The session was not started.

Action: Your program can wait until the communications configuration is varied on and then issue the acquire operation again, it can try the acquire operation again using a different device description, continue local processing, or end.

Messages:

```
CPF4285 (Escape)
CPF5333 (Escape)
```

82AD **Description:** The open or acquire operation attempted by your program to establish an SNA session was not successful because the SDLC frame size was not large enough to contain the RU size. Either this was a configuration error, or the frame size was changed to a smaller value by the Exchange ID (XID) command.

Action: End your program. The SDLC frame size is specified in the MAXFRAME parameter on the controller description, and the RU size is specified in the MAXLENRU parameter on the device description. Verify that these configuration parameters are correct and, if necessary, reduce the RU size or increase the frame size. If changes to the configuration must be made, first vary the device off, then on again. Try to run your program again.

Messages:

```
CPF4260 (Escape)
CPF5341 (Escape)
```

82B3 **Description:** The open or acquire operation attempted by your program was not successful because your program is trying to use a device description that is already in use by another job. The session was not started.

Action: Wait for the device description to become available, then issue the acquire operation again. You can use the Work with Configuration Status (WRKCFGSTS) command to determine which job is using the device description. Consider increasing the WAITFILE parameter of the CHGICFF or OVRICFF command to allow more time for the device to become available. Otherwise, your program can continue local processing or end.

Messages:

```
CPF4282 (Escape)
CPF5332 (Escape)
```

82EA **Description:** The open or acquire operation attempted by your program was not successful. A format selection of *RECID was specified on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, but cannot be used with the ICF file because the RECID DDS keyword is not used on any of the record formats in the file. The session was not started.

Action: Close the ICF file. Change the record format selection (FMTSLT) parameter to select formats by some means other than *RECID, or use a file that has a RECID DDS keyword specified for at least one record format. Open the file again.

Messages:

```
CPF4348 (Escape)
CPF5521 (Escape)
```

82EC **Description:** The acquire operation attempted by your program was not successful because finance communications does not support FMTSLT(*RMTFMT).

Action: End your program, correct the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, then run your program again.

Messages:

```
CPF4347 (Escape)
CPF5515 (Escape)
```

82EE **Description:** Your program attempted an open or acquire operation to a device that is not supported. Your program tried to acquire a device that is not a valid ICF communications type, or it is trying to acquire the requesting program device in a program that was not started by a program start request. The session was not started.

Action: Your program can continue local processing or end. Verify that the name of the remote location is specified correctly in the RMTLOCNAME parameter on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. If your program was attempting to acquire a non-ICF device, use the appropriate interface for that communications type. If your program was attempting to acquire a requesting program device, verify that your program is running in the correct environment.

Messages:

```
CPF4105 (Escape)
CPF4223 (Escape)
CPF4251 (Escape)
CPF4760 (Escape)
CPF5038 (Escape)
CPF5550 (Escape)
```

82EF **Description:** Your program attempted an acquire operation, or an open operation that implicitly acquires a session, to a device that the user is not authorized to, or that is in service mode. The session was not started.

Action: If the operation was an acquire, correct the problem and issue the acquire again. If the operation was an open, close the file, correct the problem, then issue the open operation again. To correct an authority error, obtain authority for the device from your security officer or device owner. If the device is in service mode, wait until machine service function (MSF) is no longer using the device before issuing the operation again.

Messages:

```
CPF4104 (Escape)
CPF4186 (Escape)
CPF5278 (Escape)
CPF5279 (Escape)
```

82F4 **Description:** The open or acquire operation attempted by your program was not successful because the open operation for *input only* is valid only for a requesting program device.

Action: End your program, correct the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, then run your program again.

Messages:

```
CPF4322 (Escape)
CPF5539 (Escape)
```

Major Code 83

Major Code 83 – Session error occurred (the error is recoverable).

Description: A session error occurred, but the session may still be active. Recovery within your program might be possible.

Action: You can perform the following general actions for all 83xx return codes. Specific actions are given in each minor code description.

- Correct the problem and continue processing with the session. If the error occurred because of a resource failure on the remote system or because the remote system was not active at the time, a second attempt may be successful. If the operation is still not successful, your program should end the session.
- Issue an end-of-session function and continue processing without the session.
- End.

Several of the minor codes indicate that an error condition must be corrected by changing a value in the communications configuration or in the file.

- To change a parameter value in the communications configuration, vary the configuration off, make the change to the configuration description, then vary the configuration on.
- To change a parameter value in the file, use the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.

  Note: When a parameter can be specified both in the ADDICFDEVE or OVRICFDEVE command and in the configuration, the value in the ADDICFDEVE or OVRICFDEVE command overrides the value specified in the configuration (for your program only). Therefore, in some cases, you may choose to make a change with the ADDICFDEVE or OVRICFDEVE command rather than in the configuration.

If no changes are needed in your file or in the configuration, and depending on what the return code description says, you should notify the remote location that a change is required at that location to correct the error received.

Code	Description/Action
830B	**Description:** Your program attempted an operation that was not valid because the session was not yet acquired or has ended. The session may have ended because of a release operation, an end-of-session function, or a permanent error. Your program may have incorrectly handled a previous error.

Action: Verify that your program does not attempt any operations without an active session. Also verify that your program correctly handles the permanent error or session-not-acquired return codes (80xx, 81xx, 82xx) it received on previously issued input and output operations. To recover from an incorrectly handled error condition, your program may or may not be able to issue another acquire operation, depending on the return code.

Messages:

```
CPD4079 (Diagnostic)
CPF4739 (Status)
CPF5067 (Escape)
CPF5068 (Escape)
CPF5070 (Escape)
```

8319 **Description:** The remote program sent a negative-response with sense data.

Action: Examine the sense data in the I/O feedback area to determine the necessary error recovery.

Messages:

```
CPF4813 (Notify)
CPF4814 (Notify)
```

831B **Description:** Your program tried to specify invalid sense data on a negative-response function. Correct your program so that it sends valid sense data on a negative-response function. Valid sense data must be either 0 or 8 bytes long. To send 8 bytes, the first four bytes must be 0000, 08xx, or 10xx, and the remaining four bytes must be in the ranges 0-9, A-F, or a-f. If your program chooses to send a negative-response without sense data, finance communications automatically sends 08110000 to the remote program.

Messages:

```
CPF4820 (Notify)
```

831C **Description:** Your program's previous output operation received a return code of 0412, indicating that your program must receive information sent by the remote program; however, your program did not handle the return code correctly. The current output operation was not successful because your program should have issued an input operation to receive the information already sent by the remote program.

Action: Issue an input operation to receive the previous information.

Messages:

```
CPF4934 (Notify)
```

831E **Description:** The operation attempted by your program was not valid, or a combination of operations that was not valid was specified. The session is still active. The error may have been caused by one of the following:

- Your program issued an operation that is not recognizable or not supported by finance communications.
- Your program requested a combination of operations or keywords that was not valid, such as a combined write-then-read operation with the invite function specified.
- Your program issued an input operation, or an output operation with the invite function, for a file that was opened for output only.
- Your program issued an output operation for a file that was opened for input only.
- Your program issued a close operation with a temporary close option.
- A function-management-header function was issued with zero data length.
- A function-management-header function was issued, and it was not specified in the first record of a group.

Action: Your program can try a different operation, issue a release operation or end-of-session function, or end. Correct the error in your program before trying to communicate with the remote program.

If the file was opened for input only, do not issue any output operations; or, if the file was opened for output only, do not issue any input operations, and do not use the invite function on an output operation. If such an operation is needed, then release the session, close the ICF file, and open the file again for input and output.

Messages:

```
CPF4564 (Escape)
CPF4764 (Notify)
CPF4766 (Notify)
CPF4790 (Notify)
CPF4803 (Notify)
CPF5132 (Escape)
CPF5149 (Escape)
```

831F **Description:** Your program specified data or a length for the operation that was not valid; however, the session is still active. One of the following caused the error indication:

- On an output operation, your program tried to send a data record that was longer than the MAXRCDLEN value specified for the ICF file.
- The program used a read or write operation that specified a data length greater than the record format in the ICF file.
- If this was a timer function, the format of the timer interval was not HHMMSS.
- If a system-defined format was used to specify the operation, or if the variable-length-data-record (VARLEN) function was used, then the length of the user buffer was not valid.

Action: If you want your program to recover, try the operation again with a smaller data length. If you do not need your program to recover immediately, do one of the following:

- Change the record format length in the ICF file, or change the record length in your program and compile your program again.
- For an input operation, specify a data length equal to or less than the record format length, or do not specify a length at all.
- If the timer function was used, verify that the format of the timer interval is HHMMSS.
- For an output operation that used the variable-length-data-record (VARLEN) function, verify that the length specified is less than the record length specified for the ICF file when it was opened.

Messages:

CPF4762 (Notify)
CPF4765 (Notify)
CPF4767 (Notify)

8322 **Description:** Your program tried to issue a negative-response or fail function. However, these operations are not valid at the current time.

- Your program can issue a negative-response function only when it has received data that was in error. In this case, it can issue the negative-response function on the next operation.
- Your program can issue a fail function only if it is attempting to send data or if it has received data for which an error indication can be sent.

Action: Your program can issue an output operation to continue sending data, issue an input operation to begin receiving data, issue an end-of-session function to continue local processing, or end. Correct the error that caused your program to attempt the not valid operation.

Messages:

CPF4817 (Notify)

8323 **Description:** Your program attempted to issue a cancel function when data or a negative-response indication was received for your program. The cancel function is only valid in send state.

Action: Your program can issue an input operation to continue receiving data, issue an end-of-session function, or end. Correct the error that caused your program to attempt the not valid operation.

Messages:

CPF4776 (Notify)

8326 **Description:** Your program attempted to issue a cancel function to cancel a group of records when no records were previously sent to start a group. The cancel function is only valid within a chain; it is not valid preceding a chain or between chains. The session is still active.

Action: Correct the error that caused your program to attempt the not valid operation.

Messages:

```
CPF4779 (Notify)
```

832C **Description:** A release operation following an invite function was detected. Because your program issued the invite function, it cannot issue a release operation to end the invited session.

Action: Issue an input operation to satisfy the invite function, or issue a cancel-invite function to cancel the invite function; then try the release operation again. Otherwise, issue an end-of-session function to end the session. If a coding error caused your program to attempt a release operation that was not valid, correct your program.

Messages:

```
CPF4769 (Notify)
```

832D **Description:** Following an invite function, your program issued a negative-response indication, a cancel reply, or an additional invite function. This operation failed because the original invite function must first be satisfied by an input operation.

Action: Issue an input operation to receive the data that was invited. Otherwise, issue an end-of-session function to end the session. If a coding error caused your program to attempt a request-to-write indication or an additional invite function, correct your program.

Messages:

```
CPF4924 (Notify)
```

832F **Description:** The release operation issued by your program was not successful because your program attempted the operation while the current transaction was still active, or the release operation issued by your program was not successful because of one of the following:

- The group of records sent by your program was not closed.
- Data is available for your program to receive.
- Sense data is available for your program to receive.

The release operation is not valid if a group of records was partially sent or received, or if any data or a negative-response indication were received from the remote program for which your program did not issue an input operation. The operation was not performed, but the session is still active.

Action: Use the detach function to end the current transaction before issuing a release operation. Correct the error that caused your program to issue an evoke function during an active transaction; then run your program again.

Messages:

```
CPF4819 (Notify)
CPF5099 (Notify)
```

83B6 **Description:** On an output operation, your program received an indication that the remote program has quiesced the SNA session on which this transaction is running by issuing the SNA quiesce-at-end-of-chain (QEC) command. The remote program may release the quiesced state at a later time by issuing the SNA release-quiesce command.

Action: Your program can wait and try the output operation again at a later time. Otherwise, your program can end the session, continue local processing, or end.

Messages:

```
CPF4816 (Notify)
```

83E0 **Description:** Your program attempted an operation using a record format that was not defined for the ICF file.

Action: Verify that the name of the record format in your program is correct, then check to see whether the record format is defined in the file definition.

Messages:

```
CPF5054 (Notify)
```

83E8 **Description:** Your program attempted to issue a cancel-invite function to a session that was not invited. One of the following may have occurred:

- The invite function was implicitly canceled earlier in your program by a valid output operation.
- The invite function was satisfied earlier in your program by a valid input operation.
- Your program had already canceled the invite function, then tried to cancel it again.
- Your program never invited the session.

The session is still active.

Action: Your program can issue an input or output operation, issue an end-of-session function, continue local processing, or end. However, you should correct the error that caused your program to attempt the cancel-invite to a session that was not invited.

Messages:

```
CPF4763 (Notify)
```

83F8 **Description:** Your program attempted to issue an operation to a program device that is marked in error due to a previous I/O or acquire operation. Your program may have handled the error incorrectly.

Action: Release the program device, correct the previous error, then acquire the program device again.

Messages:

```
CPF5293 (Escape)
```

Program Start Request Errors

When a program start request is rejected by the system, message CPF1269 is sent. This message contains information that can be used to determine why the program start request was rejected. Message CPF1269 is sent to the QSYSMSG message queue if that queue exists and is not damaged. If the QSYSMSG message queue is damaged or does not exist, the message is sent to the QSYSOPR message queue.

Figure B-1 shows the reason codes for rejected program start requests. This information is sent using CPF1269 to the system that issued the rejected program start request. Program start requests that request a response fail with the listed negative response sense codes; those that do not request a response fail with the listed LUSTAT sense codes.

Figure B-1 (Page 1 of 3). Reason Codes for Rejected Program Start Requests

Reason Code	Negative Response Sense Code	LUSTAT Sense Code	Description
401	084B0000	00001004	Attach request received for a device that is not allocated to an active subsystem.
402	084B0000	00001003	Requested device currently held by a Hold Communications Device (HLDCMNDEV) command.
403	084B0000	00001004	User profile cannot be accessed.
404	084B0000	00001004	Job description cannot be accessed.
405	084B0000	00001004	Output queue cannot be accessed.
406	084B0000	00001004	Maximum number of jobs defined by subsystem description are already active.
407	084B0000	00001004	Maximum number of jobs defined by communications entry are already active.
408	084B0000	00001004	Maximum number of jobs defined by routing entry are already active.
409	084B0000	00001004	Library on library list is in use exclusively by another job.
410	084B0000	00001004	Group profile cannot be accessed.
411	084B0000	00001004	Insufficient storage in machine pool to start job.
412	08120000	00001004	System values not accessible.
501	084C0000	00001000	Job description is not found.
502	084C0000	00001000	Output queue is not found.
503	084C0000	00001004	Class is not found.
504	084C0000	00001001	Library on library list is not found.
505	084C0000	00001000	Job description or job description library is damaged.
506	084C0000	00001001	Library on library list is destroyed.
507	084C0000	00001001	Duplicate libraries are found on library list.
508	084C0000	00001000	Defined size of storage pool is zero.
602	10010000	00001000	Value of transaction program name is reserved but not supported.

Figure B-1 (Page 2 of 3). Reason Codes for Rejected Program Start Requests

Reason Code	Negative Response Sense Code	LUSTAT Sense Code	Description
604	10010000	00001004	Matching routing entry is not found.
605	10010000	00001009	Program is not found.
704	080F0000	00001000	Password is not valid.
705	080F0000	00001000	User is not authorized to device.
706	080F0000	00001000	User is not authorized to subsystem description.
707	080F0000	00001000	User is not authorized to job description.
708	080F0000	00001000	User is not authorized to output queue.
709	080F0000	00001000	User is not authorized to program.
710	080F0000	00001000	User is not authorized to class.
711	080F0000	00001008	User is not authorized to library on library list.
712	080F0000	00001000	User is not authorized to group profile.
713	080F0000	00001000	User ID is not valid.
723	080F0000	00001000	There is no password associated with the user ID.
726	080F0000	00001000	User profile is disabled.
801	10010000	00001000	Program initialization parameters are present but not allowed.
802	084C0000	00001000	More than 2000 bytes of program initialization parameters received for the prestart job.
803	084B0000	00001004	Subsystem ending in progress.
804	084B0000	00001004	Prestart job is either not active or is ending.
805	084B0000	00001004	WAIT(*NO) specified on prestart job entry.
806	084B0000	00001004	MAXJOBS on prestart job entry exceeded.
807	084B0000	00001004	Prestart job ended too soon.
901	10010000	00001000	Program initialization parameters are not valid.
902	10010000	00001000	Number of parameters for program is not valid.
903	10010000	00001000	Program initialization parameters required but not sent.
1001	08640000	00001000	System logic error; function check or unexpected return code encountered.
1002	08640000	00001000	System logic error; function check or unexpected return code encountered while receiving initialization parameters.
1501	10010000	00001009	Character in procedure name not valid.
1502	10010000	00001009	Procedure not found.
1503	084C0000	00001001	System/36 environment library not found.
1504	084C0000	00001001	Library QSSP not found.
1505	084C0000	00001000	File QS36PRC not found in library QSSP.
1506	10010000	00001000	Procedure name is greater than 8 characters.
1507	084C0000	00001001	Current library not found.
1508	080F0000	00001008	Not authorized to current library.
1509	080F0000	00001000	Not authorized to file QS36PRC in current library.

Figure B-1 (Page 3 of 3). Reason Codes for Rejected Program Start Requests

Reason Code	Negative Response Sense Code	LUSTAT Sense Code	Description
1510	080F0000	00001000	Not authorized to procedure in current library.
1511	080F0000	00001008	Not authorized to System/36 environment library.
1512	080F0000	00001000	Not authorized to file QS36PRC in System/36 environment library.
1513	080F0000	00001000	Not authorized to procedure in System/36 environment library.
1514	080F0000	00001008	Not authorized to library QSSP.
1515	080F0000	00001000	Not authorized to file QS36PRC in library QSSP.
1516	080F0000	00001000	Not authorized to procedure in file QS36PRC in library QSSP.
1517	08640000	00001000	Unexpected return code from System/36 environment support.
1518	10010000	00001009	Problem phase program not found in library QSSP.
1519	080F0000	00001000	Not authorized to problem phase program in library QSSP.
1520	084B0000	00001004	Maximum number of target programs started (100 per System/36 environment).
2651	10010000	00001009	*EXEC statement not specified.
2652	10010000	00001009	Blank missing after *EXEC statement.
2653	10060002	00001009	Program name missing.
2654	10020000	00001009	Program name greater than 10 characters.
2655	10020000	00001001	Library name greater than 10 characters.

Appendix C. Mapping Intersystem Communications Function Operations to Systems Network Architecture Commands

This appendix shows the association between some of the Systems Network Architecture (SNA) commands and ICF finance communications application operations.

In Figure C-1, the ICF operation appears with a corresponding interaction between the AS/400 system and the finance controller with a finance device varied on. The device is varied on before this scenario begins.

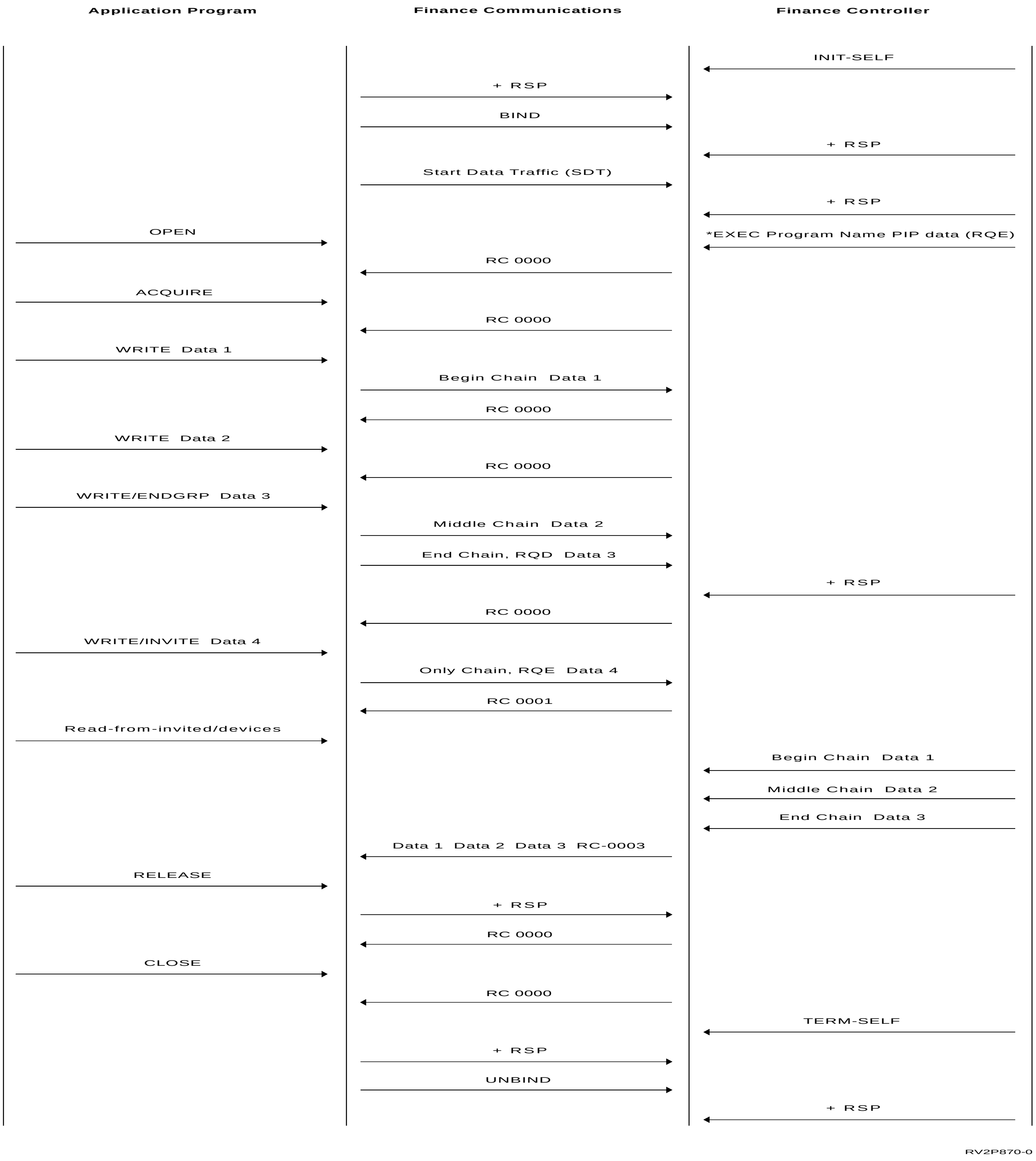


Figure C-1. SNA Commands with Corresponding ICF Application Operations

In Figure C-2, the SNA session is initiated by a controller application program.

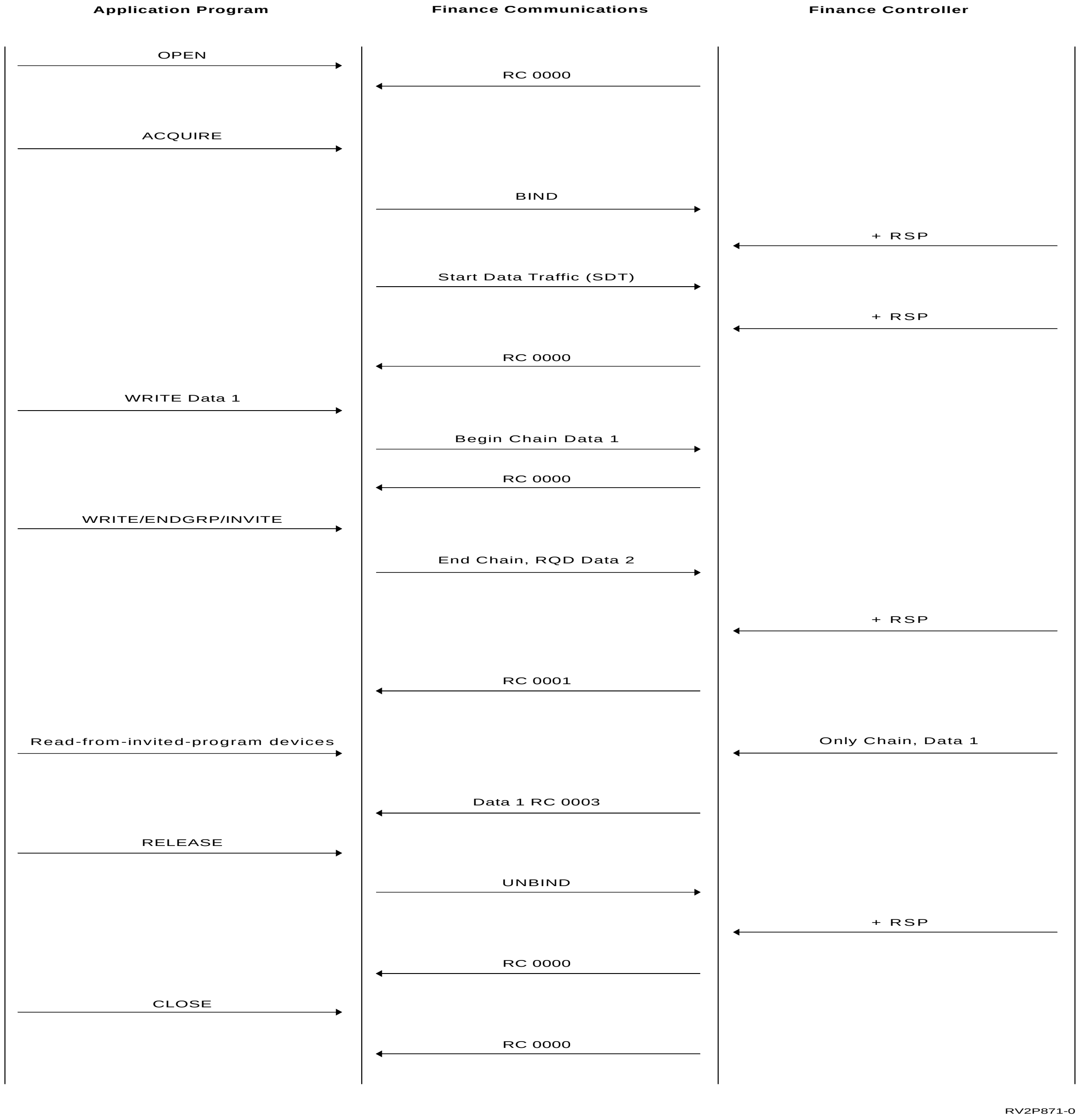


Figure C-2. SNA Commands with Corresponding ICF Application Operations on an AS/400 System-Initiated SNA Session

Appendix D. 4701 Finance Controller Diskette Download

A 4701 finance controller must have an operating or initial program-load (IPL) diskette before it can be used. This diskette contains the basic license internal code, the controller configuration, and the application programs that can be created on a System/370 computer configured to support the 4700 finance communications system. The operating image created by the System/370 computer is loaded on the AS/400 system, and the file is sent to the 4701 controller using the system monitor session. This appendix describes the support that sends the file to the controller after it is loaded on the system.

When the 4701 controller receives the file, the controller uses the file to create an operational diskette that it later uses during its own IPL procedure. The procedure for sending an operational diskette image is usually done only for the following actions:

- Installing the 4701 controller
- Changing the 4701 application programs
- Upgrading the 4701 controller

Using the Send Finance Diskette Image Command

The download support can be accessed through the Send Finance Diskette Image (SNDFNCIMG) command. This command includes the following information:

The SNDFNCIMG command uses the file, member, and remote location name.

File
Specifies the library and the file where the diskette image resides.

Remote location name
Specifies the remote location name of a finance device specified as TYPE(*FNCICF). This device must be attached to a 4701 finance controller with an 8-inch diskette drive or a 3601 controller configured as a 4701 controller. The local location address (the address of the logical unit) of the device *must* be 01.

Member
Specifies the member in the file containing the diskette image which was blocked into a basic exchange format.

Have the person responsible for security authorize your user profile to the SNDFNCIMG command and the QCRFDWNLD ICF file. Use the Grant Object Authority (GRTOBJAUT) to grant authority.

```
GRTOBJAUT OBJ(QSYS/SNDFNCIMG) OBJTYPE(*CMD)
          USER(user-name)  AUT(*CHANGE)

GRTOBJAUT OBJ(QSYS/QCRFDWNLD) OBJTYPE(*FILE)
          USER(user-name)  AUT(*CHANGE)
```

To send the image, do the following:

1. Do an IPL of the controller with the operating diskette you are currently using.
2. Prepare the controller to accept the image. Start the system monitor on the controller and issue the 999 command to create a diskette.
3. Vary on the system monitor to be used.

   Note: All devices attached to the controller description must be varied off before the system monitor device is varied on.
4. Enter the SNDFNCIMG command.

```
                                                                   Job: I  Pgm: I  Exec

              ┌─*LIBL/────────┐
►►──SNDFNCIMG──FILE(─┼───────────────┼─file-name──)─────────────────────────────────►
              ├─*CURLIB/──────┤
              └─library-name/─┘

►──RMTLOCNAME(──remote-location-name──)─(P)─┬──────────────────────────────┬──────►◄
                                            │         ┌─*FIRST──────┐      │
                                            └─MBR(────┴─member-name─┴──)───┘
```

Note:
P All parameters preceding this point can be specified positionally.

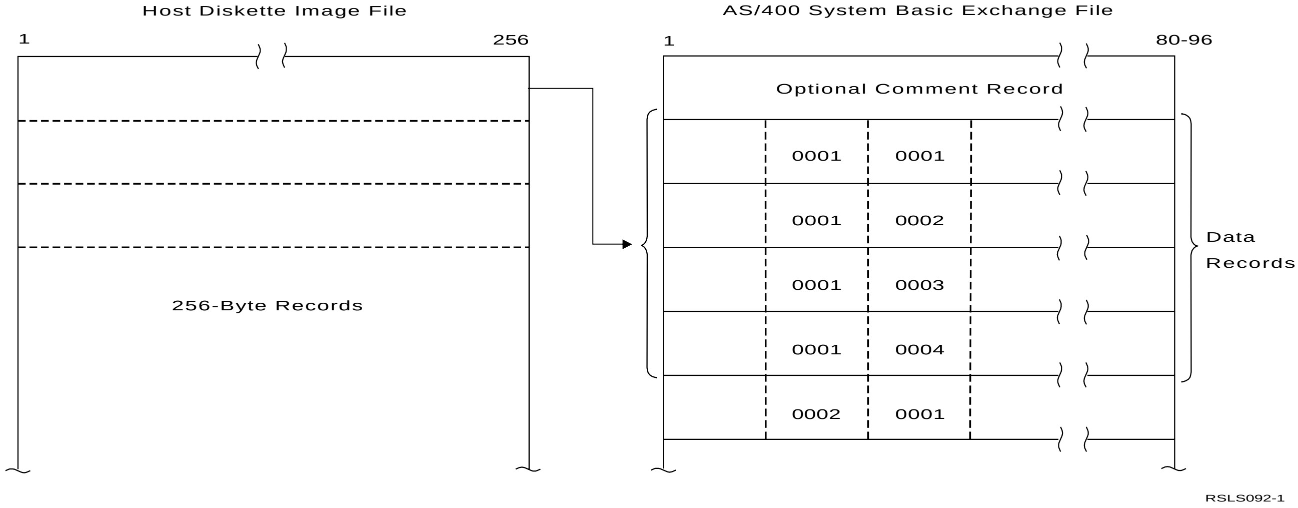


Figure D-1. Diskette Image Format and Basic Exchange Files

The SNDFNCIMG command creates the original image again and sends the operating image to the controller. The controller builds the operating diskette by writing the operating image on a blank diskette. When this is done, you can use the diskette for the IPL procedure for the controller.

Figure D-1 shows the basic format of the diskette image and the basic exchange files.

If you cannot get the basic exchange file, you can create the file with the following procedure:

1. Create a diskette image file using the Host Diskette Image Create (HDIC) program. This program is a part of 4700 Finance Communications System Host Support for an IBM System/370 computer, 3031, 3032, 3033, or 4300. The diskette image file must be converted into a basic exchange file that can be sent by the SNDFNCIMG command.

   The first record in the file is an optional comment record. You can write any information in this record to identify the file. The remaining records contain the data from the diskette image file.

2. Convert each 256-byte record from the diskette image into four 64-byte records, shown in Figure D-1.

3. Write the records to the basic exchange file. The required format for each data record is shown in Figure D-2 on page D-3.

   Each record can be from 80 to 96 bytes in length. You can use positions 1 through 8 for an optional header or comments. You can use positions 80 through 96, if needed, for optional comments.

   The SNDFNCIMG command uses the sequential block number and sequential record number to ensure the correct sequence when the command processes the file.

The data field contains 64 bytes of data from the diskette image file.

For more information about the diskette downloading support, see the *IBM 4700 Finance Communications System: Subsystem Operating Procedures*, and the *Host Support User's Guide*.

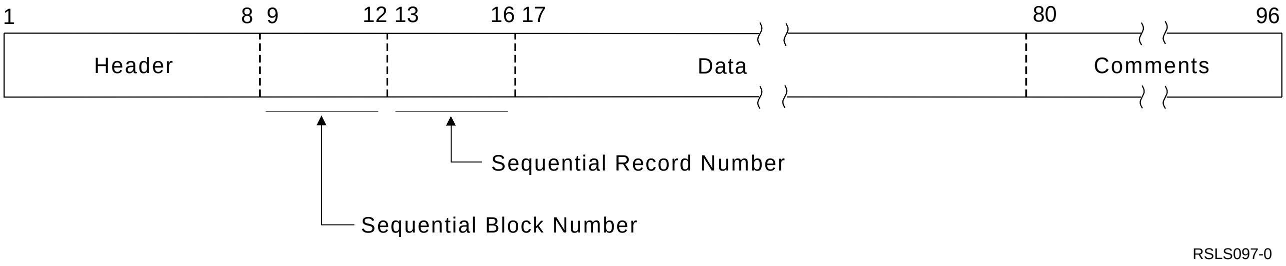


Figure D-2. Format Required for Data Records

Appendix E. Intersystem Communications Function Finance Example Programs

This appendix provides examples of COBOL/400, RPG/400, and ILE C/400 programs to demonstrate how finance communications is used.

Figure E-1 on page E-2 shows representations of example programs.

COBOL/400 Source Program for Local System

This section describes the objects needed on the local system to run the ILE COBOL/400 account inquiry finance program.

Configuration

The following configuration commands are used to create the synchronous data link control (SDLC) line, controller, and device descriptions used by the local system.

```
CRTLINSDLC LIND(FNCLINE) RSRCNAME(LIN022) ONLINE(*NO)
           ROLE(*PRI) NRZI(*YES)
CRTCTLFNC  CTLD(FNCCTL) TYPE(4702) MODEL(0) LINKTYPE(*SDLC)
           ONLINE(*NO) LINE(FNCLINE) STNADR(01)
CRTDEVFNC  DEVD(K001DEV) TYPE(*FNCICF) LOCADR(03)
           RMTLOCNAME(K001DEV) ONLINE(*NO) CTL(FNCCTL)
```

Program Files

The following files are used by the local system.

K001ICF
The ICF file used to send and receive records from the finance controller. This file was created by using the following command:

```
CRTICFF FILE(FNCLIB/K001ICF) SRCFILE(FNCLIB/QDDSSRC)
        SRCMBR(K001ICF)
```

The following command defines the program device entry:

```
ADDICFDEVE FILE(FNCLIB/K001ICF) PGMDEV(FNCTRGT)
           RMTLOCNAME(*REQUESTER)
```

An OVRICFDEVE command with the same parameters can also be used.

K001DBF
The database file that holds the account records. This file was created by using the following command:

```
CRTPF FILE(FNCLIB/K001DBF) SRCFILE(FNCLIB/QDDSSRC)
      SRCMBR(K001DBF)
```

K001PRT
The printer file used to format output to a printer. This file was created by using the following command:

```
CRTPRTF FILE(FNCLIB/K001PRT) SRCFILE(FNCLIB/QDDSSRC)
        SRCMBR(K001PRT)
```

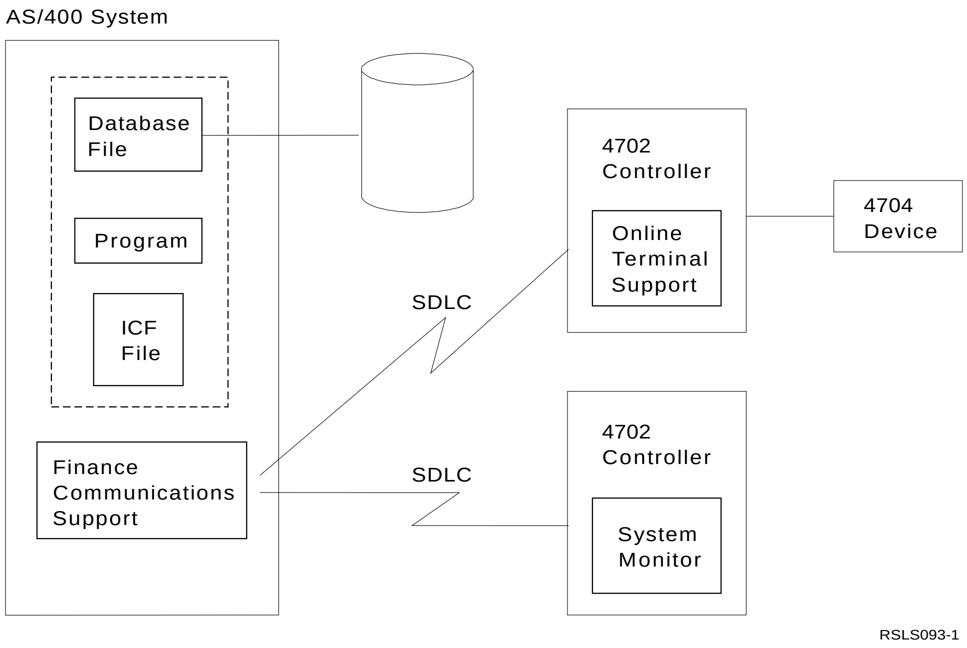


Figure E-1. Example for ICF Finance Network

```
File name . . . . . . . . . . . . . . . . . . . . . :  K001ICF
  Library name  . . . . . . . . . . . . . . . . . . :  FNCLIB
File attribute  . . . . . . . . . . . . . . . . . . :  ICF
Source file containing DDS  . . . . . . . . . . . . :  QDDSSRC
  Library name  . . . . . . . . . . . . . . . . . . :  FNCLIB
Source member containing DDS  . . . . . . . . . . . :  K001ICF
Source member last changed  . . . . . . . . . . . . :  06/01/89  17:21:35
Source listing options  . . . . . . . . . . . . . . :  *SOURCE    *LIST      *NOSECLVL
DDS generation severity level . . . . . . . . . . . :  20
DDS flagging severity level . . . . . . . . . . . . :  00
Authority . . . . . . . . . . . . . . . . . . . . . :  *LIBCRTAUT
Replace file  . . . . . . . . . . . . . . . . . . . :  *YES
Text  . . . . . . . . . . . . . . . . . . . . . . . :
Compiler  . . . . . . . . . . . . . . . . . . . . . :  IBM AS/400 Data Description Processor
                                     Data Description Source
SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date
   100       A************************************************************
   200       A*                                                          *
   300       A*                     DDS                                  *
   400       A*                FOR THE ICF FILE                          *
   500       A*     USED IN ACCOUNT INQUIRY APPLICATION PROGRAM          *
   600       A*                                                          *
   700       A************************************************************
   800       A*
   900       A* FILE LEVEL INDICATORS:
  1000       A*
  1100       A                                      INDARA
  1200       A*
  1300       A************************************************************
  1400       A*                 RETAIL RECORD FORMATS                    *
  1500       A************************************************************
  1600       A          R VARREC                    VARLEN(&LENREC)
  1700       A                                      INVITE                               06/01/89
  1800       A            DTAREC       512A
  1900       A            LENREC         5S  P
```

Figure E-2 (Part 1 of 2). DDS Source for ICF File K001ICF

```
                         * * * * *  E N D   O F   S O U R C E   * * * * *
                                          Expanded Source
                                                                                          Field       Buffer position
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
   1100                                                INDARA
   1600                    R VARREC                    VARLEN(&LENREC) INVITE
   1800                      DTAREC       512A  B                                           512           1       1
   1900                      LENREC         5S 0P                                             5         513
                 * * * * *  E N D   O F   E X P A N D E D   S O U R C E   * * * * *
                                          Message Summary
    Total          Informational          Warning        Error       Severe
                       (0-9)              (10-19)       (20-29)      (30-99)
        0                0                    0            0            0
* CPC7301      00              Message . . . . :   File K001ICF created in library FNCLIB.
                         * * * * *  E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure E-2 (Part 2 of 2). DDS Source for ICF File K001ICF

```
 File name . . . . . . . . . . . . . . . . . . . . . . :   K001DBF
   Library name  . . . . . . . . . . . . . . . . . . . :   FNCLIB
 File attribute  . . . . . . . . . . . . . . . . . . . :   Physical
 Source file containing DDS  . . . . . . . . . . . . . :   QDDSSRC
   Library name  . . . . . . . . . . . . . . . . . . . :   FNCLIB
 Source member containing DDS  . . . . . . . . . . . . :   K001DBF
 Source member last changed  . . . . . . . . . . . . . :   05/31/89  11:05:08
 Source listing options  . . . . . . . . . . . . . . . :   *SOURCE    *LIST      *NOSECLVL
 DDS generation severity level . . . . . . . . . . . . :   20
 DDS flagging severity level . . . . . . . . . . . . . :   00
 File type . . . . . . . . . . . . . . . . . . . . . . :   *DATA
 Authority . . . . . . . . . . . . . . . . . . . . . . :   *LIBCRTAUT
 Replace file  . . . . . . . . . . . . . . . . . . . . :   *NO
 Text  . . . . . . . . . . . . . . . . . . . . . . . . :
 Compiler  . . . . . . . . . . . . . . . . . . . . . . :   IBM AS/400 Data Description Processor
                                      Data Description Source
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date

     10      A**********************************************************
     20      A*                                                        *
     30      A*                     DDS                                *
     40      A*            FOR THE DATABASE FILE                       *
     50      A*     USED IN ACCOUNT INQUIRY APPLICATION PROGRAM        *
     60      A*                                                        *
     70      A**********************************************************
     80      A*
     90      A                                    UNIQUE
    100      A          R ACCOUNTR
    110      A            ACCTNR         8  0
    120      A            NAME          21
    130      A            STR1           3
    140      A            STR2          18
    150      A            CITY          16
    160      A            ZIP            5
    170      A            OCUP          21
    180      A            TBAL          10  2
    190      A            DLYWTH        10  2
    200      A            DLYDEP        10  2
    210      A            LIMIT1        10  2
    220      A            LIMIT2        10  2
    230      A            RSVCS1        10  2
    240      A            RSVCS2        10  2
    250      A            RSVRM1        10  2
    260      A            RSVRM2        10  2
    270      A            ACTIVE         1  0
    280      A            LCKDSC        38
    290      A            WTHDRL        10  2
    300      A          K ACCTNR
                        * * * * *   E N D   O F   S O U R C E   * * * * *
```

Figure E-3 (Part 1 of 2). DDS Source for Database File K001DBF

```
                                          Expanded Source
                                                                                           Field      Buffer position
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
     90                                   UNIQUE
    100                 R ACCOUNTR
    110                   ACCTNR         8P 0B      COLHDG('ACCTNR')                        5           1       1
    120                   NAME          21A  B      COLHDG('NAME')                         21           6       6
    130                   STR1           3A  B      COLHDG('STR1')                          3          27      27
    140                   STR2          18A  B      COLHDG('STR2')                         18          30      30
    150                   CITY          16A  B      COLHDG('CITY')                         16          48      48
    160                   ZIP            5A  B      COLHDG('ZIP')                           5          64      64
    170                   OCUP          21A  B      COLHDG('OCUP')                         21          69      69
    180                   TBAL          10P 2B      COLHDG('TBAL')                          6          90      90
    190                   DLYWTH        10P 2B      COLHDG('DLYWTH')                        6          96      96
    200                   DLYDEP        10P 2B      COLHDG('DLYDEP')                        6         102     102
    210                   LIMIT1        10P 2B      COLHDG('LIMIT1')                        6         108     108
    220                   LIMIT2        10P 2B      COLHDG('LIMIT2')                        6         114     114
    230                   RSVCS1        10P 2B      COLHDG('RSVCS1')                        6         120     120
    240                   RSVCS2        10P 2B      COLHDG('RSVCS2')                        6         126     126
    250                   RSVRM1        10P 2B      COLHDG('RSVRM1')                        6         132     132
    260                   RSVRM2        10P 2B      COLHDG('RSVRM2')                        6         138     138
    270                   ACTIVE         1P 0B      COLHDG('ACTIVE')                        1         144     144
    280                   LCKDSC        38A  B      COLHDG('LCKDSC')                       38         145     145
    290                   WTHDRL        10P 2B      COLHDG('WTHDRL')                        6         183     183
    300                 K ACCTNR
                * * * * *   E N D   O F   E X P A N D E D   S O U R C E   * * * * *
                                          Message Summary
    Total          Informational          Warning        Error       Severe
                        (0-9)              (10-19)       (20-29)      (30-99)
        0               0                     0             0            0
* CPC7301      00              Message . . . . :   File K001DBF created in library FNCLIB.
                        * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure E-3 (Part 2 of 2). DDS Source for Database File K001DBF

```
 File name . . . . . . . . . . . . . . . . . . . . . . :   K001PRT
   Library name  . . . . . . . . . . . . . . . . . . . :   FNCLIB
 File attribute  . . . . . . . . . . . . . . . . . . . :   Printer
 Source file containing DDS  . . . . . . . . . . . . . :   QDDSSRC
   Library name  . . . . . . . . . . . . . . . . . . . :   FNCLIB
 Source member containing DDS  . . . . . . . . . . . . :   K001PRT
 Source member last changed  . . . . . . . . . . . . . :   06/01/89  20:24:18
 Source listing options  . . . . . . . . . . . . . . . :   *SOURCE    *LIST      *NOSECLVL
 DDS generation severity level . . . . . . . . . . . . :   20
 DDS flagging severity level . . . . . . . . . . . . . :   00
 Authority . . . . . . . . . . . . . . . . . . . . . . :   *LIBCRTAUT
 Replace file  . . . . . . . . . . . . . . . . . . . . :   *YES
 Text  . . . . . . . . . . . . . . . . . . . . . . . . :
 Compiler  . . . . . . . . . . . . . . . . . . . . . . :   IBM AS/400 Data Description Processor
                                      Data Description Source
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date
     10      A************************************************************
     20      A*                                                          *
     30      A*                      DDS                                 *
     40      A*              FOR THE PRINTER FILE                        *
     50      A*     USED IN ACCOUNT INQUIRY APPLICATION PROGRAM          *
     60      A*                                                          *
     70      A************************************************************
     80      A*
     90      A************************************************************
    100      A*                  RECORD FORMATS                          *
    110      A************************************************************
    120      A          R ERRREC                   SPACEB(3)
    130      A                                    5'PROGRAM TERMINATED ABNORMALLY'
    140      A                                    5'PROGRAM DEVICE:' SPACEB(2)
    150      A            PGMDEV        10       +1
    160      A                                    5'RECORD FORMAT:' SPACEB(2)
    170      A            FMTNM          8       +1
    180      A                                    5'MAJOR CODE:' SPACEB(2)
    190      A            MAJOR          2       +1
    200      A                                    5'MINOR CODE:' SPACEB(2)
    210      A            MINOR          2       +1
                        * * * * *   E N D   O F   S O U R C E   * * * * *
                                          Expanded Source
                                                                                         Field       Buffer position
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
    120                 R ERRREC                   SPACEB(3)
    130                                           5'PROGRAM TERMINATED ABNORMALLY'          29
    140                                           5'PROGRAM DEVICE:' SPACEB(2)              15
    150                   PGMDEV        10A  O    21                                        10            1
    160                                           5'RECORD FORMAT:' SPACEB(2)               14
    170                   FMTNM          8A  O    20                                         8           11
    180                                           5'MAJOR CODE:' SPACEB(2)                  11
    190                   MAJOR          2A  O    17                                         2           19
    200                                           5'MINOR CODE:' SPACEB(2)                  11
    210                   MINOR          2A  O    17                                         2           21
                * * * * *   E N D   O F   E X P A N D E D   S O U R C E   * * * * *
                                          Message Summary
    Total          Informational          Warning        Error       Severe
                       (0-9)              (10-19)       (20-29)      (30-99)
        0                0                    0             0           0
* CPC7301      00              Message . . . . :   File K001PRT created in library FNCLIB.
                        * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure E-4. DDS Source for Printer File K001PRT

Program Explanation: The following explains the COBOL/400 account inquiry program example in Figure E-5 on page E-6.

1 This section of the program defines the database file (K001DBF), the ICF file (K001ICF), and the printer file (K001PRT) used in the program.

K001DBF is the database file that contains the customer account information.

K001ICF is the ICF file that sends records to and receives records from the online terminal support (OTS) application program on the finance controller.

K001PRT is the printer file that sends communication error information to a printer device from the account inquiry program on the AS/400 system.

2 The ERROR-SECTION section of the program defines the error handling procedure for I/O errors on the K001ICF file. It is automatically called when an exception occurs while the program is running. Feedback data is moved to a printer record (ERRREC) and the record is printed. The session is ended by releasing the finance device. The files are then closed and the program is ended.

3 The program opens the files to be used. The program device (FNCTRGT) used by the program is explicitly acquired. This program device was previously added to the ICF file (K001ICF) by the ADDICFDEVE command.

4 The parameters passed to the account inquiry program by the finance controller are placed in a data structure. If the account number is not among the parameters passed, a message is sent to the finance controller and the program is ended.

5 A read operation is done from the database file (with the account number as the key).

6 If the account number is not in the database file, a message is sent to the finance controller and the program is ended. Otherwise, the account inquiry is a valid inquiry, and control is passed to 7 to process the transaction.

7 This routine sends the inquiry reply in two records. The first record contains customer information, and the second record contains account information.

Note: The finance controller supports an RU size of 256 bytes. Therefore, if the inquiry reply was to be sent using one record, the data separators (hex 'FFFFFFFF') in the customer information data structure (FORMATTED-DATA-2) must be removed, and the 16-byte finance header must be inserted again after the first 256 bytes.

8 This routine is called to build and send the error messages to the finance controller by passing control to 11.

9 This routine is called to build and send the customer information to the finance controller by passing control to 11.

10 This routine is called to build and send the account information to the finance controller by passing control to 11.

11 A write operation with the account inquiry response is sent to the program device that sent the inquiry.

Note: For performance reasons, the INVITE keyword is specified as part of VARREC format. For more information, see “Responses” on page 7-1.

12 This routine is called to end the program. The session is ended by releasing the finance device and closing the files.

```
Program  . . . . . . . . . . . . . . . :   K001
  Library  . . . . . . . . . . . . . . :     FNCLIB
Source file  . . . . . . . . . . . . . :   PGMSRC
  Library  . . . . . . . . . . . . . . :     FNCLIB
Source member  . . . . . . . . . . . . :   K001     07/17/89 14:12:37
Generation severity level  . . . . . . :   29
Text 'description' . . . . . . . . . . :   *BLANK
Source listing options . . . . . . . . :   *NONE
Generation options . . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . . :   29
Print file . . . . . . . . . . . . . . :   QSYSPRT
  Library  . . . . . . . . . . . . . . :     *LIBL
FIPS flagging  . . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . . :   *NOFLAG
Flagging severity  . . . . . . . . . . :   0
Replace program  . . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . . :   *USER
Authority  . . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
    1  000100 IDENTIFICATION DIVISION.
       000200
    2  000300 PROGRAM-ID.  K001.
       000400
       000500*************************************************************
       000600*                                                           *
       000700*     PROGRAM TO HANDLE ACCOUNT INQUIRY FROM A FINANCE      *
       000800*     CONTROLLER.                                           *
       000900*                                                           *
       001000*************************************************************
       001100
    3  001200 ENVIRONMENT DIVISION.
       001300
    4  001400 CONFIGURATION SECTION.
    5  001500 SOURCE-COMPUTER.  IBM-AS400.
    6  001600 OBJECT-COMPUTER.  IBM-AS400.
       001700
    7  001800 SPECIAL-NAMES.    I-O-FEEDBACK IS FEEDBACK-AREA.
       001900
    8  002000 INPUT-OUTPUT SECTION.
1   9  002100 FILE-CONTROL.
       002200
   10  002300     SELECT K001DBF
   11  002400          ASSIGN TO DATABASE-K001DBF
   12  002500          ORGANIZATION IS INDEXED
   13  002600          ACCESS MODE IS RANDOM
   14  002700          RECORD KEY IS EXTERNALLY-DESCRIBED-KEY.
       002800
   15  002900     SELECT K001ICF
   16  003000          ASSIGN TO WORKSTATION-K001ICF-SI
   17  003100          ORGANIZATION IS TRANSACTION
   18  003200          CONTROL-AREA IS TRAN-CTL-AREA
   19  003300          FILE STATUS IS STATUS-IND MAJ-MIN.
       003400
   20  003500     SELECT K001PRT
   21  003600          ASSIGN TO FORMATFILE-K001PRT
   22  003700          ORGANIZATION IS SEQUENTIAL.
       003800
   23  003900 DATA DIVISION.
       004000
```

Figure E-5 (Part 1 of 11). COBOL/400 Program

```
  24  004100 FILE SECTION.
      004200
  25  004300 FD  K001DBF
  26  004400     LABEL RECORDS ARE STANDARD.
  27  004500 01  ACCOUNT-REC.
  28  004600     COPY DDS-ACCOUNTR-I-O OF K001DBF.
     +000001*    I-O FORMAT:ACCOUNTR   FROM FILE K001DBF    OF LIBRARY FNCLIB             ACCOUNTR
     +000002*                                                                             ACCOUNTR
     +000003*THE KEY DEFINITIONS FOR RECORD FORMAT  ACCOUNTR                              ACCOUNTR
     +000004*  NUMBER               NAME               RETRIEVAL     TYPE    ALTSEQ       ACCOUNTR
     +000005*   0001   ACCTNR                           ASCENDING   SIGNED     NO         ACCOUNTR
  29 +000006       05  ACCOUNTR.                                                          ACCOUNTR
  30 +000007           06 ACCTNR                PIC S9(8)        COMP-3.                  ACCOUNTR
  31 +000008           06 NAME                  PIC X(21).                                ACCOUNTR
  32 +000009           06 STR1                  PIC X(3).                                 ACCOUNTR
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  33 +000010           06 STR2                  PIC X(18).                                ACCOUNTR
  34 +000011           06 CITY                  PIC X(16).                                ACCOUNTR
  35 +000012           06 ZIP                   PIC X(5).                                 ACCOUNTR
  36 +000013           06 OCUP                  PIC X(21).                                ACCOUNTR
  37 +000014           06 TBAL                  PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
  38 +000015           06 DLYWTH                PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
  39 +000016           06 DLYDEP                PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
  40 +000017           06 LIMIT1                PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
  41 +000018           06 LIMIT2                PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
  42 +000019           06 RSVCS1                PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
  43 +000020           06 RSVCS2                PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
  44 +000021           06 RSVRM1                PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
  45 +000022           06 RSVRM2                PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
  46 +000023           06 ACTIVE                PIC S9(1)        COMP-3.                  ACCOUNTR
  47 +000024           06 LCKDSC                PIC X(38).                                ACCOUNTR
  48 +000025           06 WTHDRL                PIC S9(8)V9(2)   COMP-3.                  ACCOUNTR
      004700
  49  004800 FD  K001ICF
  50  004900     LABEL RECORDS ARE STANDARD.
  51  005000 01  ICFREC.  COPY DDS-ALL-FORMATS OF K001ICF.
  52 +000001       05  K001ICF-RECORD PIC X(517).                                         <-ALL-FMTS
     +000002*  INPUT FORMAT:VARREC     FROM FILE K001ICF    OF LIBRARY FNCLIB             <-ALL-FMTS
     +000003*                                                                             <-ALL-FMTS
  53 +000004       05  VARREC-I      REDEFINES K001ICF-RECORD.                            <-ALL-FMTS
  54 +000005           06 DTAREC                PIC X(512).                               <-ALL-FMTS
     +000006* OUTPUT FORMAT:VARREC     FROM FILE K001ICF    OF LIBRARY FNCLIB             <-ALL-FMTS
     +000007*                                                                             <-ALL-FMTS
  55 +000008       05  VARREC-O      REDEFINES K001ICF-RECORD.                            <-ALL-FMTS
  56 +000009           06 DTAREC                PIC X(512).                               <-ALL-FMTS
  57 +000010           06 LENREC                PIC S9(5).                                <-ALL-FMTS
      005100
  58  005200 FD  K001PRT
  59  005300     DATA RECORD IS PRT-REC.
  60  005400 01  PRT-REC. COPY DDS-ALL-FORMATS-O OF K001PRT.
  61 +000001       05  K001PRT-RECORD PIC X(22).                                          <-ALL-FMTS
     +000002* OUTPUT FORMAT:ERRREC     FROM FILE K001PRT    OF LIBRARYFNCLIB             <-ALL-FMTS
     +000003*                                                                             <-ALL-FMTS
  62 +000004       05  ERRREC-O      REDEFINES K001PRT-RECORD.                            <-ALL-FMTS
  63 +000005           06 PGMDEV                PIC X(10).                                <-ALL-FMTS
  64 +000006           06 FMTNM                 PIC X(8).                                 <-ALL-FMTS
  65 +000007           06 MAJOR                 PIC X(2).                                 <-ALL-FMTS
  66 +000008           06 MINOR                 PIC X(2).                                 <-ALL-FMTS
      005500
  67  005600 WORKING-STORAGE SECTION.
      005700
  68  005800 77  STATUS-IND      PIC XX.
  69  005900 77  INDON           PIC 1 VALUE B"1".
  70  006000 77  INDOFF          PIC 1 VALUE B"0".
      006100**************************************************************
      006200*  PROGRAM DEVICES                                          *
      006300**************************************************************
  71  006400 77  WS-PGMDEV       PIC X(10) VALUE "FNCTRGT   ".
      006500
  72  006600 01  PGM-INDIC-AREA.
  73  006700     05 PGM-INDIC    PIC 1 OCCURS 99 TIMES
```

Figure E-5 (Part 2 of 11). COBOL/400 Program

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  74  006800                     INDICATOR 1.
      006900
  75  007000 01  TRAN-CTL-AREA.
  76  007100     05  FILLER        PIC X(2).
  77  007200     05  PGM-DEV-NAME  PIC X(10).
  78  007300     05  RCD-FMT-NAME  PIC X(10).
      007400
  79  007500 01  FEEDBACK-DATA.
  80  007600     05  FILLER        PIC X(20).                                                      07/17/89
  81  007700     05  FMTNM         PIC X(10).
  82  007800     05  FILLER        PIC X(2).                                                       07/17/89
  83  007900     05  PGMDEV        PIC X(10).
  84  008000     05  FILLER        PIC X(136).                                                     07/17/89
  85  008100     05  MAJOR         PIC X(2).
  86  008200     05  MINOR         PIC X(2).
      008300
  87  008400 01  MAJ-MIN.
  88  008500     05  MAJ           PIC X(2).
  89  008600     05  MIN           PIC X(2).
      008700
  90  008800 01  FILLER.
  91  008900     05    FFFFFF PIC 9(8) COMP-4 VALUE 16777215.
  92  009000     05    FILLER REDEFINES FFFFFF.
  93  009100          10  FILLER PIC X.
  94  009200          10  FFS    PIC XXX.
      009300
      009400*************************************************************
      009500* FORMAT OF THE DATA SENT BY THE FINANCE CONTROLLER.        *
      009600*************************************************************
  95  009700 01       FNC-INPUT-DATA.
  96  009800     10      FNC-CTL-ITEMS.
  97  009900          15       WSTYPE         PIC XX.
  98  010000          15       CTLUNIT        PIC XX.
  99  010100          15       WSNO           PIC XX.
 100  010200          15       AUDTNO         PIC XX.
 101  010300          15       TELLERNO       PIC 999.
 102  010400          15       LINENBR        PIC 99.
 103  010500          15       TRNCDE         PIC XXX.
 104  010600          15       SPLFNCT        PIC X.
 105  010700          15       RESVRD         PIC XXX.
 106  010800          15       CTLUTYPE       PIC X.
 107  010900     10     FNC-DATA-AREA.
 108  011000          15       FLD1P          PIC X.
 109  011100          15       FLD2P          PIC X.
 110  011200          15       FLD3P          PIC X.
 111  011300          15       FLD4P          PIC X.
 112  011400          15       FLD5P          PIC X.
 113  011500          15       FLD6P          PIC X.
 114  011600          15       FLD7P          PIC X.
 115  011700          15       INPUT-FIELD    PIC X(10) OCCURS 7 TIMES
 116  011800                                             INDEXED BY FLD.
      011900
      012000*************************************************************
      012100* FORMAT OF THE DATA SENT BY THE AS/400 TO THE FINANCE      *
      012200* CONTROLLER.                                               *
      012300*************************************************************
```

Figure E-5 (Part 3 of 11). COBOL/400 Program

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
 117  012400 01  FNC-OUTPUT-DATA.
 118  012500     05     FNC-CONTROL.
      012600*************************************************************
      012700* THE TERMINAL-MODE VARIABLE PUTS THE 4704 DISPLAY INTO     *
      012800* LARGE SCREEN MODE WHEREBY 1920 CHARACTERS ARE DISPLAYED,  *
      012900* THEREBY ALLOWING MORE INFORMATION TO BE DISPLAYED, WHEN   *
      013000* THE VARIABLE IS SET TO 1.                                 *
      013100*************************************************************
 119  013200          10        TERMINAL-MODE  PIC 9     VALUE IS 0.
 120  013300          10        FILLER         PIC X(4)  VALUE SPACES.
      013400*************************************************************
      013500* THE MORE-DATA VARIABLE INFORMS THE FINANCE CONTROLLER THAT*
      013600* MORE DATA IS YET TO COME (THE TRANSACTION HAS NOT         *
      013700* COMPLETED), WHEN THE VARIABLE IS SET TO 1.                *
      013800*************************************************************
 121  013900          10        MORE-DATA      PIC 9     VALUE IS 0.
 122  014000          10        FILLER         PIC X(10) VALUE SPACES.
 123  014100     05     FNC-FORMATTED-DATA     PIC X(240).
      014200
      014300*************************************************************
      014400* DATA FORMAT FOR ERROR MESSAGE DISPLAY. THE FOLLOWING      *
      014500* ERRORS ARE HANDLED: ACCOUNT NUMBER NOT FOUND.             *
      014700*************************************************************
 124  014800 01  FORMATTED-DATA-1.
 125  014900     05   NEW-PAGE-1     PIC X.
 126  015000     05   TEXT-1         PIC X(30).
 127  015100     05   ACCTNR         PIC ZZZZZZZZ.
 128  015200     05   DATA-SEP-1     PIC X.
 129  015300     05   LASTFF-1       PIC XXX.
      015400
      015500*************************************************************
      015600* DATA FORMAT FOR FIRST HALF OF VALID INQUIRY DISPLAY.      *
      015700*************************************************************
 130  015800 01  FORMATTED-DATA-2.
 131  015900     05   NEW-PAGE-2     PIC X.
 132  016000     05   SETPOS-2A      PIC X.
 133  016100     05   MOVHOR-2A      PIC X.
 134  016200     05   HEX20-2A       PIC X.
 135  016300     05   DATE-2         PIC ZZ/ZZ/ZZ.
 136  016400     05   NEWLIN-2B      PIC X.
 137  016500     05   FILLER         PIC X(5) VALUE "ACC#:".
 138  016600     05   ACCTNR         PIC ZZZZZZZ9.
 139  016700     05   FILLER         PIC X.
 140  016800     05   NAME           PIC X(21).
 141  016900     05   NEWLIN-2C      PIC X.
 142  017000     05   SETPOS-2C      PIC X.
 143  017100     05   MOVHOR-2C      PIC X.
 144  017200     05   HEX0F-2C       PIC X.
 145  017300     05   STR1           PIC XXX.
 146  017400     05   STR2           PIC X(18).
 147  017500     05   NEWLIN-2D      PIC X.
 148  017600     05   SETPOS-2D      PIC X.
 149  017700     05   MOVHOR-2D      PIC X.
```

Figure E-5 (Part 4 of 11). COBOL/400 Program

```
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  150  017800     05   HEX0F-2D       PIC X.
  151  017900     05   CITY           PIC X(16).
  152  018000     05   NEWLIN-2E      PIC X.
  153  018100     05   SETPOS-2E      PIC X.
  154  018200     05   MOVHOR-2E      PIC X.
  155  018300     05   HEX0F-2E       PIC X.
  156  018400     05   OCUP           PIC X(21).
  157  018500     05   DATA-SEP-2     PIC X.
  158  018600     05   LASTFF-2       PIC XXX.
       018700
       018800*************************************************************
       018900* DATA FORMAT FOR SECOND HALF OF VALID INQUIRY DISPLAY.     *
       019000*************************************************************
  159  019100 01  FORMATTED-DATA-3.
  160  019200     05   SETPOS-3       PIC X.
  161  019300     05   SETLIN-3       PIC X.
  162  019400     05   HEX06-3        PIC X.
  163  019500     05   FILLER         PIC X(5) VALUE "BAL :".
  164  019600     05   TBAL           PIC ZZ,ZZZ,ZZZ.99-.
  165  019700     05   FILLER         PIC X(5).
  166  019800     05   FILLER         PIC X(10) VALUE "TOT.RSRVD:".
  167  019900     05   NEWLIN-3A      PIC X.
  168  020000     05   FILLER         PIC X(5) VALUE "DEP :".
  169  020100     05   DLYDEP         PIC ZZ,ZZZ,ZZZ.99.
  170  020200     05   FILLER         PIC X(3).
  171  020300     05   TOTRSV-3       PIC ZZ,ZZZ,ZZZ.99-.
  172  020400     05   NEWLIN-3B      PIC X.
  173  020500     05   FILLER         PIC X(5) VALUE "WTH :".
  174  020600     05   DLYWTH         PIC ZZ,ZZZ,ZZZ.99.
  175  020700     05   NEWLIN-3C      PIC X.
  176  020800     05   FILLER         PIC X(5) VALUE "LMT1:".
  177  020900     05   LIMIT1         PIC ZZ,ZZZ,ZZZ.99.
  178  021000     05   NEWLIN-3D      PIC X.
  179  021100     05   FILLER         PIC X(5) VALUE "LMT2:".
  180  021200     05   LIMIT2         PIC ZZ,ZZZ,ZZZ.99.
  181  021300     05   NEWLIN-3E      PIC X.
  182  021400     05   FILLER         PIC X(5) VALUE "W/B :".
  183  021500     05   WRKBAL-3       PIC ZZ,ZZZ,ZZZ.99.
  184  021600     05   NEWLIN-3F      PIC X.
  185  021700     05   LCKDSC         PIC X(38).
  186  021800     05   DATA-SEP-3     PIC X.
  187  021900     05   LASTFF-3       PIC XXX.
       022000
       022100*************************************************************
       022200* DATA AREA TO DEFINE THE HEX VALUES NEEDED IN THE PROGRAM  *
       022300* TO SEND CONTROL CHARACTERS TO THE FINANCE CONTROLLER.     *
       022400*************************************************************
  188  022500 01  DUMMY1.
       022600*    05   HEX0C          PIC X.
       022700*    05   HEX15          PIC X.
  189  022800     06   BIN1           PIC 9999 COMP-4 VALUE IS 3093.
       022900*    05   HEX08          PIC X.
       023000*    05   HEX34          PIC X.
  190  023100     06   BIN2           PIC 9999 COMP-4 VALUE IS 2100.
       023200*    05   HEX04          PIC X.
```

Figure E-5 (Part 5 of 11). COBOL/400 Program

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      023300*    05   HEXFF          PIC X.
 191  023400     06   BIN3           PIC 9999 COMP-4 VALUE IS 1279.
      023500*    05   HEX02          PIC X.
      023600*    05   HEX12          PIC X.
 192  023700     06   BIN4           PIC 9999 COMP-4 VALUE IS 530.
      023800*    05   HEX0D          PIC X.
      023900*    05   HEX25          PIC X.
 193  024000     06   BIN5           PIC 9999 COMP-4 VALUE IS 3365.
      024100*    05   HEX0F          PIC X.
      024200*    05   HEX20          PIC X.
 194  024300     06   BIN6           PIC 9999 COMP-4 VALUE IS 3872.
      024400*    05   HEX0A          PIC X.
      024500*    05   HEX06          PIC X.
 195  024600     06   BIN7           PIC 9999 COMP-4 VALUE IS 2566.
      024700*    05   DUMMY00        PIC X.
      024800*    05   LINENBR-HEX    PIC X.
 196  024900     06   BIN-LINENBR    PIC 9999 COMP-4.
      025000
      025100*************************************************************
      025200* REDEFINES AREA TO GIVE THE HEX CONTROL CODES MEANINGFUL   *
      025300* NAMES.                                                    *
      025400*************************************************************
 197  025500 01  DUMMY2                             REDEFINES DUMMY1.
 198  025600     05   NEWPAG         PIC X.
 199  025700     05   NEWLIN         PIC X.
 200  025800     05   MOVHOR         PIC X.
 201  025900     05   SETPOS         PIC X.
 202  026000     05   SETLIN         PIC X.
 203  026100     05   DTASEP         PIC X.
 204  026200     05   HEX02          PIC X.
 205  026300     05   HEX12          PIC X.
 206  026400     05   HEX0D          PIC X.
 207  026500     05   HEX25          PIC X.
 208  026600     05   HEX0F          PIC X.
 209  026700     05   HEX20          PIC X.
 210  026800     05   HEX0A          PIC X.
 211  026900     05   HEX06          PIC X.
 212  027000     05   DUMMY00        PIC X.
 213  027100     05   LINENBR-HEX    PIC X.
      027200
      027300*************************************************************
      027400* THESE ARE THE MESSAGES USED BY THE PROGRAM.               *
      027500*************************************************************
 214  027600 01  MESSAGES.
 215  027700   05   MSG1      PIC X(30) VALUE "ACCOUNT NR. NOT FOUND".
 216  027800   05   MSG2      PIC X(30) VALUE "ACCOUNT NR. NOT PASSED".
      027900
 217  028000 01  ACCOUNT-STATUS PIC 9.
 218  028100     88   VALID-ACCOUNT VALUE IS 1.
      028200
 219  028300 01  EDATE          PIC  999999.
 220  028400 01  ACCTNO         PIC  S9(8).
 221  028500 01  WRKBAL         PIC  9(8)V99  COMP-3.
 222  028600 01  TOTRSV         PIC  9(8)V99  COMP-3.
      028700
```

Figure E-5 (Part 6 of 11). COBOL/400 Program

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      028800*************************************************************
      028900* THESE ARE THE PARAMETERS PASSED TO THE PROGRAM BY THE     *
      029000* FINANCE SUPPORT FACILITY.                                 *
      029100*************************************************************
 223  029200 LINKAGE SECTION.
 224  029300 01  DATA-PARM      PIC X(256).
 225  029400 PROCEDURE DIVISION USING DATA-PARM.
      029500
      029600**************************************************************
      029700*                                                            *
      029800*  THE FOLLOWING DECLARATIVES SECTION IS AN ERROR ROUTINE    *
      029900*  THAT IS RUN WHEN AN ERROR OCCURS ON THE READ OR WRITE     *
      030000*  OF THE ICF FILE "K001ICF". THE ROUTINE MOVES DATA FROM    *
      030100*  THE I-O FEEDBACK AREA TO THE "ERROR" FORMAT OF THE PRINT  *
      030200*  FILE "K001PRT". THE PROGRAM IS TERMINATED RELEASING       *
      030300*  THE FINANCE DEVICE, AND CLOSING ALL FILES.                *
      030400*                                                            *
      030500**************************************************************
2     030600 DECLARATIVES.
      030700 ERROR-SECTION SECTION.
      030800     USE AFTER EXCEPTION PROCEDURE ON I-O.
      030900 ERROR-PARAGRAPH.
 226  031000     ACCEPT FEEDBACK-DATA FROM FEEDBACK-AREA.
 227  031100     MOVE CORRESPONDING FEEDBACK-DATA TO ERRREC-O.
            *       ** CORRESPONDING items for statement 227:
            *       **     FMTNM
            *       **     PGMDEV
            *       **     MAJOR
            *       **     MINOR
            *       ** End of CORRESPONDING items for statement 227
 228  031200     WRITE PRT-REC FORMAT IS "ERRREC".
      031300
 229  031400     DROP WS-PGMDEV FROM K001ICF.
      031500
 230  031600     CLOSE K001ICF
      031700           K001DBF
      031800           K001PRT.
      031900
 231  032000     STOP RUN.
      032100 END DECLARATIVES.
      032200
      032300 MAIN-PROGRAM SECTION.
3     032400 INITIALIZE-PROGRAM.
 232  032500     OPEN I-O     K001ICF.
 233  032600     OPEN I-O     K001DBF.
 234  032700     OPEN OUTPUT  K001PRT.
      032800
 235  032900     ACQUIRE WS-PGMDEV  FOR K001ICF.
      033000
 236  033100     ACCEPT EDATE FROM DATE.
 237  033200     MOVE ZEROES TO PGM-INDIC-AREA.
      033300
      033400*************************************************************
      033500*                                                           *
      033600* THE DATA COMES INTO THE PROGRAM FROM THE FINANCE          *
```

Figure E-5 (Part 7 of 11). COBOL/400 Program

```
  STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
        033700* CONTROLLER WITH UP TO SEVEN FIELDS. THERE ARE SEVEN FLAGS *
        033800* THAT INDICATE WHICH FIELDS ARE PRESENT:                   *
        033900*    IF FIELD ONE IS PRESENT, FLD1P IS "1", OTHERWISE       *
        034000*    IT IS " ".                                             *
        034100*    IF FIELD TWO IS PRESENT, FLD2P IS "2", OTHERWISE       *
        034200*    IT IS " ".                                             *
        034300*    AND SO ON . . .                                        *
        034400* THE DATA IS LOADED SEQUENTIALLY INTO THE AVAILABLE FIELDS *
        034500* SO IF THE OPERATOR ENTERS FIELDS 1, 3, 5, AND 7, THE DATA *
        034600* WILL BE STORED IN INPUT FIELDS 1, 2, 3, AND 4. IT IS      *
        034700* NECESSARY TO TEST FOR THE PRESENCE OF THE FIELDS AND MOVE *
        034800* EACH INPUT FIELD INTO THE CORRECT FIELD IN THE PROGRAM.   *
        034900* SINCE THIS PROGRAM ONLY DEALS WITH ACCOUNT INQUIRIES, IF  *
        035000* AN ACCOUNT NUMBER IS NOT PASSED, A MESSAGE WILL BE SENT.  *
        035100*                                                           *
        035200*************************************************************
        035300
   238  035400     MOVE DATA-PARM TO FNC-INPUT-DATA.
4  239  035500     SET FLD TO 1.
        035600
   240  035700     IF FLD1P = "1"
        035800     THEN
   241  035900       MOVE INPUT-FIELD(FLD) TO ACCTNO
   242  036000       PERFORM PROCESS-TRANSACTION
        036100     ELSE
   243  036200       MOVE MSG2 TO TEXT-1
   244  036300       PERFORM BUILD-FORMAT-1.
        036400
   245  036500     PERFORM CLEAN-UP.
        036600
        036700
        036800*************************************************************
        036900*                                                           *
        037000* BEGIN PROCESSING THE TRANSACTION -                        *
        037100* A READ FROM THE DATABASE FILE IS DONE USING THE ACCOUNT   *
        037200* NUMBER AS THE KEY.  IF THE READ WAS SUCCESSFUL, CUSTOMER  *
        037300* AND ACCOUNT INFORMATION IS SENT BACK TO THE CONTROLLER.   *
        037400* IF THE READ WAS UNSUCCESSFUL, A MESSAGE STATING THAT THE  *
        037500* ACCOUNT NUMBER WAS NOT FOUND WILL BE SENT TO THE          *
        037600* CONTROLLER. THE FILES ARE THEN CLOSED AND THE PROGRAM     *
        037700* IS TERMINATED.                                            *
        037800*                                                           *
        037900*************************************************************
        038000 PROCESS-TRANSACTION.
   246  038100     MOVE 1 TO ACCOUNT-STATUS.
5  247  038200     MOVE ACCTNO TO ACCTNR OF ACCOUNT-REC.
   248  038300     READ K001DBF INVALID KEY MOVE 0 TO ACCOUNT-STATUS.
        038400
6  250  038500     IF VALID-ACCOUNT
        038600     THEN
   251  038700       PERFORM VALID-INQUIRY
        038800     ELSE
   252  038900       MOVE MSG1 TO TEXT-1
   253  039000       PERFORM BUILD-FORMAT-1.
        039100
```

Figure E-5 (Part 8 of 11). COBOL/400 Program

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      039200
      039300**************************************************************
      039400*                                                            *
      039500* THE PROCEDURE TO DO A VALID INQUIRY SENDS OUT THE DISPLAY *
      039600* IN TWO RECORDS. THE FIRST RECORD HAS THE BASIC CUSTOMER   *
      039700* INFORMATION ACCOUNT, NAME, AND ADDRESS.  THE SECOND HAS   *
      039800* THE BALANCE INFORMATION.                                  *
      039900*                                                            *
      040000**************************************************************
 7    040100 VALID-INQUIRY.
 254  040200     PERFORM BUILD-FORMAT-2.
 255  040300     COMPUTE WRKBAL = TBAL   OF ACCOUNT-REC
      040400             - DLYWTH OF ACCOUNT-REC
      040500             + DLYDEP OF ACCOUNT-REC
      040600             + LIMIT1 OF ACCOUNT-REC
      040700             + LIMIT2 OF ACCOUNT-REC.
 256  040800     COMPUTE TOTRSV = RSVCS1 + RSVCS2 + RSVRM1 + RSVRM2.
 257  040900     PERFORM BUILD-FORMAT-3.
      041000
      041100**************************************************************
      041200*                                                            *
      041300* A MESSAGE IS SENT TO THE CONTROLLER.                       *
      041400*                                                            *
      041500**************************************************************
 8    041600 BUILD-FORMAT-1.
 258  041700     MOVE ACCTNO TO ACCTNR OF FORMATTED-DATA-1.
 259  041800     MOVE 0 TO TERMINAL-MODE.
 260  041900     MOVE NEWPAG TO NEW-PAGE-1.
 261  042000     MOVE DTASEP TO DATA-SEP-1.
 262  042100     MOVE FFS TO LASTFF-1
      042200
 263  042300     MOVE FORMATTED-DATA-1 TO FNC-FORMATTED-DATA.
      042400
 264  042500     MOVE 59 TO LENREC OF VARREC-O.
 265  042600     PERFORM SEND-RESPONSE.
      042700
      042800**************************************************************
      042900*                                                            *
      043000* THE CUSTOMER INFORMATION IS SENT TO THE CONTROLLER.        *
      043100*                                                            *
      043200**************************************************************
 9    043300 BUILD-FORMAT-2.
 266  043400     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-2.
            *       ** CORRESPONDING items for statement 266:
            *       **     ACCTNR
            *       **     NAME
            *       **     STR1
            *       **     STR2
            *       **     CITY
            *       **     OCUP
            *       ** End of CORRESPONDING items for statement 266
 267  043500     MOVE EDATE TO DATE-2.
      043600
 268  043700     MOVE 1 TO TERMINAL-MODE.
 269  043800     MOVE NEWPAG TO NEW-PAGE-2.
```

Figure E-5 (Part 9 of 11). COBOL/400 Program

```
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  270  043900     MOVE NEWLIN TO NEWLIN-2B, NEWLIN-2C, NEWLIN-2D, NEWLIN-2E.
  271  044000     MOVE SETPOS TO SETPOS-2A, SETPOS-2C, SETPOS-2D, SETPOS-2E.
  272  044100     MOVE MOVHOR TO MOVHOR-2A, MOVHOR-2C, MOVHOR-2C, MOVHOR-2E.
  273  044200     MOVE HEX20 TO HEX20-2A.
  274  044300     MOVE HEX0F TO HEX0F-2C, HEX0F-2D, HEX0F-2E.
  275  044400     MOVE DTASEP TO DATA-SEP-2.
  276  044500     MOVE FFS TO LASTFF-2
       044600
  277  044700     MOVE 1 TO MORE-DATA.
       044800
  278  044900     MOVE FORMATTED-DATA-2 TO FNC-FORMATTED-DATA.
       045000
  279  045100     MOVE 138 TO LENREC OF VARREC-O.
  280  045200     PERFORM SEND-RESPONSE.
       045300
       045400*************************************************************
       045500*                                                           *
       045600* THE ACCOUNT INFORMATION IS SENT TO THE CONTROLLER.        *
       045700*                                                           *
       045800*************************************************************
 10    045900 BUILD-FORMAT-3.
  281  046000     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-3.
            *       ** CORRESPONDING items for statement 281:
            *       **     TBAL
            *       **     DLYWTH
            *       **     DLYDEP
            *       **     LIMIT1
            *       **     LIMIT2
            *       **     LCKDSC
            *       ** End of CORRESPONDING items for statement 281
  282  046100     MOVE WRKBAL TO WRKBAL-3.
  283  046200     MOVE TOTRSV TO TOTRSV-3.
  284  046300     MOVE 1 TO TERMINAL-MODE.
  285  046400     MOVE NEWLIN TO NEWLIN-3A, NEWLIN-3B, NEWLIN-3C, NEWLIN-3D,
       046500             NEWLIN-3E, NEWLIN-3F.
  286  046600     MOVE SETPOS TO SETPOS-3.
  287  046700     MOVE SETLIN TO SETLIN-3.
  288  046800     MOVE HEX06 TO HEX06-3
  289  046900     MOVE DTASEP TO DATA-SEP-3.
  290  047000     MOVE FFS TO LASTFF-3
       047100
  291  047200     MOVE 0 TO MORE-DATA.
  292  047300     MOVE FORMATTED-DATA-3 TO FNC-FORMATTED-DATA.
       047400
  293  047500     MOVE 208 TO LENREC OF VARREC-O.
  294  047600     PERFORM SEND-RESPONSE.
       047700
       047800*************************************************************
       047900*                                                           *
       048000* RESPOND TO THE ACCOUNT INQUIRY.                           *
       048100*                                                           *
       048200*************************************************************
 11    048300 SEND-RESPONSE.
  295  048400     MOVE FNC-OUTPUT-DATA TO DTAREC OF VARREC-O.
       048500
```

Figure E-5 (Part 10 of 11). COBOL/400 Program

```
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  296  048600     WRITE ICFREC
       048700       FORMAT IS "VARREC"
       048800       TERMINAL IS WS-PGMDEV.
       048900
       049000*************************************************************
       049100*                                                           *
       049200* TERMINATE PROGRAM.                                        *
       049300*                                                           *
       049400*************************************************************
 12    049500 CLEAN-UP.
  297  049600     DROP WS-PGMDEV FROM K001ICF.
       049700
  298  049800     CLOSE K001ICF
       049900           K001DBF
       050000           K001PRT.
       050100
  299  050200     STOP RUN.
       050300
                         * * * * *   E N D   O F   S O U R C E   * * * * *
 STMT
                     * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                       Message Summary
 Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
     0         0            0               0               0                0
Source records read . . . . . . . . :   503
Copy records read . . . . . . . . . :   43
Copy members processed  . . . . . . :   3
Sequence errors . . . . . . . . . . :   0
Highest severity message issued . . :   0
 LBL0901 00  Program K001 created in library FNCLIB.
                    * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure E-5 (Part 11 of 11). COBOL/400 Program

RPG/400 Source Program for Local System

This section describes the objects needed on the local system to run the ILE RPG/400 CPGEN download finance program. See Figure E-9 on page E-21.

Configuration

This section lists the configuration commands used to create the synchronous data link control (SDLC) line, controller, and device descriptions used by the local system.

```
CRTLINSDLC LIND(LSYSMON) RSRCNAME(LIN022) ONLINE(*NO)
           ROLE(*PRI) NRZI(*YES)
CRTCTLFNC  CTLD(CSYSMON) TYPE(4702) MODEL(0)
           LINKTYPE(*SDLC) ONLINE(*NO) LINE(LSYSMON)
           STNADR(C1)
CRTDEVFNC  DEVD(CPGDEV) TYPE(*FNCICF) LOCADR(01)
           RMTLOCNAME(CPGDEV) ONLINE(*NO) CTL(CSYSMON)
```

Program Files

The following files are used by the local system.

CPGICF

The ICF file used to send and receive records from the finance controller. This file was created by using the following command:

```
CRTICFF FILE(FNCLIB/CPGICF) SRCFILE(FNCLIB/QDDSSRC)
        SRCMBR(CPGICF)
```

The following command defines the program device entry:

```
ADDICFDEVE FILE(FNCLIB/CPGICF) PGMDEV(PGMDEV)
           RMTLOCNAME(CPGDEV)
```

An OVRICFDEVE command with the same parameters can also be used.

CPGDBF

The database file used to hold the records of the CPGEN file that is to be downloaded. This file was created by using the following command:

```
CRTPF FILE(FNCLIB/CPGDBF) SRCFILE(FNCLIB/QDDSSRC)
      SRCMBR(CPGDBF)
```

CPGPRT

The printer file used to format output to a printer. This file was created by using the following command:

```
CRTPRTF FILE(FNCLIB/CPGPRT) SRCFILE(FNCLIB/QDDSSRC)
        SRCMBR(CPGPRT)
```

```
File name . . . . . . . . . . . . . . . . . . . . . :   CPGICF
  Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
File attribute  . . . . . . . . . . . . . . . . . . :   ICF
Source file containing DDS  . . . . . . . . . . . . :   QDDSSRC
  Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
Source member containing DDS  . . . . . . . . . . . :   CPGICF
Source member last changed  . . . . . . . . . . . . :   06/05/89   9:11:40
Source listing options  . . . . . . . . . . . . . . :   *SOURCE    *LIST      *NOSECLVL
DDS generation severity level . . . . . . . . . . . :   20
DDS flagging severity level . . . . . . . . . . . . :   00
Authority . . . . . . . . . . . . . . . . . . . . . :   *LIBCRTAUT
Replace file  . . . . . . . . . . . . . . . . . . . :   *YES
Text  . . . . . . . . . . . . . . . . . . . . . . . :
Compiler  . . . . . . . . . . . . . . . . . . . . . :   IBM AS/400 Data Description Processor
                                     Data Description Source
SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date
   100       A***********************************************************
   200       A*                                                         *
   300       A*                      DDS                                *
   400       A*                FOR THE ICF FILE                         *
   500       A*     USED IN CPGEN FILE DOWNLOAD APPLICATION PROGRAM     *               06/05/89
   600       A*                                                         *
   700       A***********************************************************
   800       A*
   900       A* FILE LEVEL INDICATORS:
  1000       A*
  1100       A
  1200       A                                      INDARA
  1300       A*
  1400       A***********************************************************
  1500       A*                 RETAIL RECORD FORMATS                   *
  1600       A***********************************************************
  1700       A          R DATAREC
  1800       A            DATA         256A
  1900       A          R DATAEND                   ENDGRP
  2000       A            DATA         256A
  2100       A          R INIT                      ENDGRP
  2200       A            DATA2          2A
  2300       A          R TERM                      ENDGRP
  2400       A                                      INVITE
  2500       A            DATA2          2A
                        * * * * *  E N D  O F  S O U R C E  * * * * *
                                         Expanded Source
                                                                                         Field       Buffer position
SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
  1200                                              INDARA
  1700                  R DATAREC
  1800                    DATA         256A  B                                             256           1       1
  1900                  R DATAEND                   ENDGRP
  2000                    DATA         256A  B                                             256           1       1
  2100                  R INIT                      ENDGRP
  2200                    DATA2          2A  B                                               2           1       1
  2300                  R TERM                      ENDGRP INVITE
  2500                    DATA2          2A  B                                               2           1       1
                * * * * *  E N D  O F  E X P A N D E D  S O U R C E  * * * * *
                                         Message Summary
   Total          Informational          Warning        Error       Severe
                      (0-9)              (10-19)       (20-29)      (30-99)
       0                 0                    0             0            0
* CPC7301      00               Message . . . . :   File CPGICF created in library FNCLIB.
                        * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure E-6. DDS Source for ICF File CPGICF

```
 File name . . . . . . . . . . . . . . . . . . . . . . :   CPGDBF
   Library name  . . . . . . . . . . . . . . . . . . . :   FNCLIB
 File attribute  . . . . . . . . . . . . . . . . . . . :   Physical
 Source file containing DDS  . . . . . . . . . . . . . :   QDDSSRC
   Library name  . . . . . . . . . . . . . . . . . . . :   FNCLIB
 Source member containing DDS  . . . . . . . . . . . . :   CPGDBF
 Source member last changed  . . . . . . . . . . . . . :   06/05/89   9:10:57
 Source listing options  . . . . . . . . . . . . . . . :   *SOURCE    *LIST      *NOSECLVL
 DDS generation severity level . . . . . . . . . . . . :   20
 DDS flagging severity level . . . . . . . . . . . . . :   00
 File type . . . . . . . . . . . . . . . . . . . . . . :   *DATA
 Authority . . . . . . . . . . . . . . . . . . . . . . :   *LIBCRTAUT
 Replace file  . . . . . . . . . . . . . . . . . . . . :   *NO
 Text  . . . . . . . . . . . . . . . . . . . . . . . . :
 Compiler  . . . . . . . . . . . . . . . . . . . . . . :   IBM AS/400 Data Description Processor
                                       Data Description Source
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date
     100      A***********************************************************
     200      A*                                                         *
     300      A*                      DDS                                *
     400      A*             FOR THE DATABASE FILE                       *
     500      A*     USED IN CPGEN FILE DOWNLOAD APPLICATION PROGRAM     *                06/05/89
     600      A*                                                         *
     700      A***********************************************************
     800      A*
     900      A          R CPGREC
    1000      A            CPG           80A
                         * * * * *  E N D  O F  S O U R C E  * * * * *
                                          Expanded Source
                                                                                          Field       Buffer position
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
     900                 R CPGREC
    1000                   CPG           80A  B      COLHDG('CPG')                           80            1       1
                 * * * * *  E N D  O F  E X P A N D E D  S O U R C E  * * * * *
                                          Message Summary
    Total          Informational          Warning        Error       Severe
                       (0-9)              (10-19)       (20-29)      (30-99)
        0                 0                    0             0           0
* CPC7301      00               Message . . . . :   File CPGDBF created in library FNCLIB.
                         * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure   E-7. DDS Source for Database File CPGDBF

```
File name . . . . . . . . . . . . . . . . . . . . . :   CPGPRT
  Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
File attribute  . . . . . . . . . . . . . . . . . . :   Printer
Source file containing DDS  . . . . . . . . . . . . :   QDDSSRC
  Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
Source member containing DDS  . . . . . . . . . . . :   CPGPRT
Source member last changed  . . . . . . . . . . . . :   06/05/89   9:12:28
Source listing options  . . . . . . . . . . . . . . :   *SOURCE    *LIST      *NOSECLVL
DDS generation severity level . . . . . . . . . . . :   20
DDS flagging severity level . . . . . . . . . . . . :   00
Authority . . . . . . . . . . . . . . . . . . . . . :   *LIBCRTAUT
Replace file  . . . . . . . . . . . . . . . . . . . :   *YES
Text  . . . . . . . . . . . . . . . . . . . . . . . :
Compiler  . . . . . . . . . . . . . . . . . . . . . :   IBM AS/400 Data Description Processor
                                     Data Description Source
SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date
   100       A***********************************************************
   200       A*                                                         *
   300       A*                      DDS                                *
   400       A*               FOR THE PRINTER FILE                      *
   500       A*    USED IN CPGEN FILE DOWNLOAD APPLICATION PROGARAM     *                06/05/89
   600       A*                                                         *
   700       A***********************************************************
   800       A*
   900       A***********************************************************
  1000       A*                   RECORD FORMATS                        *
  1100       A***********************************************************
  1200       A          R ERRREC                    SPACEB(3)
  1300       A                                     5'PROGRAM TERMINATED ABNORMALLY'
  1400       A                                     5'PROGRAM DEVICE:' SPACEB(2)
  1500       A            PGMDEV        10        +1
  1600       A                                     5'RECORD FORMAT:' SPACEB(2)
  1700       A            FMTNM          8        +1
  1800       A                                     5'MAJOR CODE:' SPACEB(2)
  1900       A            MAJOR          2        +1
  2000       A                                     5'MINOR CODE:' SPACEB(2)
  2100       A            MINOR          2        +1
  2200       A          R RSPERR                    SPACEB(3)
  2300       A                                     5'UNSUCCESSFUL OPERATION DUE '
  2400       A                                     5'TO BAD STATUS CODE.' SPACEB(2)
                        * * * * *   E N D   O F   S O U R C E   * * * * *
                                         Expanded Source
                                                                                         Field       Buffer position
SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
  1200                  R ERRREC                    SPACEB(3)
  1300                                             5'PROGRAM TERMINATED ABNORMALLY'         29
  1400                                             5'PROGRAM DEVICE:' SPACEB(2)             15
  1500                    PGMDEV        10A  O    21                                        10            1
  1600                                             5'RECORD FORMAT:' SPACEB(2)              14
  1700                    FMTNM          8A  O    20                                         8           11
  1800                                             5'MAJOR CODE:' SPACEB(2)                 11
  1900                    MAJOR          2A  O    17                                         2           19
  2000                                             5'MINOR CODE:' SPACEB(2)                 11
  2100                    MINOR          2A  O    17                                         2           21
  2200                  R RSPERR                    SPACEB(3)
  2300                                             5'UNSUCCESSFUL OPERATION DUE '           27
  2400                                             5'TO BAD STATUS CODE.' SPACEB(2)         19
                * * * * *   E N D   O F   E X P A N D E D   S O U R C E   * * * * *
                                         Message Summary
   Total          Informational          Warning        Error       Severe
                      (0-9)              (10-19)       (20-29)      (30-99)
       0                 0                    0             0            0
* CPC7301      00               Message . . . . :   File CPGPRT created in library FNCLIB.
                        * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure E-8. DDS Source for Printer File CPGPRT

Program Explanation: This section explains the ILE RPG/400 CPGEN file download program example in Figure E-9 on page E-21.

1 This section of the program defines the database file (CPGDBF), the ICF file (CPGICF), and the printer file (CPGPRT) used in the program.

CPGDBF is the database file that contains the CPGEN file to be downloaded.

CPGICF is the ICF file that sends records to and receives records from the system monitor program on the finance controller.

CPGPRT is the printer file that sends communications error information to a printer device from the CPGEN file download program on the AS/400 system.

Note: The files used in this program are opened at the beginning of the ILE RPG/400 cycle.

2 FEEDBK is the name of the file information data structure (INFDS) used with the CPGICF file. The FEEDBK data structure contains the following information:

- Record format name (FMTNM)
- Program device name (PGMDEV)
- Major/Minor return code (MAJOR, MINOR)

3 WRKBUF is the name of the structure that holds records to be sent to the finance controller and that holds records received from the finance controller.

4 CNSTDS is the name of the structure used by the program for building the initialization and termination sequences that are sent to the system monitor program on the finance controller. The initialization sequence tells the system monitor program that a CPGEN file is coming. The termination sequence tells the system monitor program that the CPGEN file download program on the AS/400 system has finished sending data.

5 FILL00 is the name of the structure used by the program to set records to hex '00'.

6 This section explicitly acquires the program device (PGMDEV) used by the program. This program device was previously added to the ICF file (CPGICF) by the ADDICFDEVE command.

7 This section builds and sends the initialization sequence to the finance controller.

8 Data is read from the database file (CPGDBF) and sent to the finance controller until all records in the database file are sent (the indicator 99 is set on).

9 This section of the program sends the termination sequence to the finance controller.

10 The system monitor program on the finance controller sends a response back to the termination sequence sent in 9. If the response received is hex '0581', the CPGEN file download was successful and the indicator 95 is set; otherwise, control is passed to 13 and a message is printed. Then the program is ended.

11 This section of the program does the end-of-job processing. First, the session with the system monitor program on the finance controller is ended by releasing the finance device. Because additional processing is not needed in this program, the LR indicator is set on and all files are closed implicitly. Then the program ends.

12 This subroutine builds the initialization and termination sequences sent to the finance controller and builds constants used throughout the program.

13 When a response is received that indicates the records sent to the system monitor program on the finance controller were rejected, this subroutine prints a printer record (RSPERR) that indicates the CPGEN file download was not successful.

14 This subroutine is automatically called when an exception occurs while the program is running. A printer record (ERRREC) is printed with information regarding the error condition. The *CANCL option on the ENDSR operation causes the program to end and all files to close.

```
Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . . . . . . . . :   FNCLIB/CPGPGM
  Source file  . . . . . . . . . . :   FNCLIB/PGMSRC
  Source member  . . . . . . . . . :   CPGPGM
  Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
  Source listing indentation . . . :   *NONE
  SAA flagging . . . . . . . . . . :   *NOFLAG
  Generation severity level  . . . :   9
  Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
  Replace program  . . . . . . . . :   *YES
  Target release . . . . . . . . . :   *CURRENT
  User profile . . . . . . . . . . :   *USER
  Authority  . . . . . . . . . . . :   *LIBCRTAUT
  Text . . . . . . . . . . . . . . :   *SRCMBRTXT
  Phase trace  . . . . . . . . . . :   *NO
  Intermediate text dump . . . . . :   *NONE
  Snap dump  . . . . . . . . . . . :   *NONE
  Codelist . . . . . . . . . . . . :   *NONE
  Ignore decimal data error  . . . :   *NO
Actual Program Source:
  Member . . . . . . . . . . . . . :   CPGPGM
  File . . . . . . . . . . . . . . :   PGMSRC
  Library  . . . . . . . . . . . . :   FNCLIB
  Last Change  . . . . . . . . . . :   06/03/89  12:05:45
SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
                         S o u r c e   L i s t i n g
1         H                                                                                              *****
     100  FCPGDBF  IF  E           K        DISK
           RECORD FORMAT(S):  LIBRARY FNCLIB FILE CPGDBF.
                    EXTERNAL FORMAT CPGREC RPG NAME CPGREC
     200  FCPGPRT  O   E            66     PRINTER
           RECORD FORMAT(S):  LIBRARY FNCLIB FILE CPGPRT.
                    EXTERNAL FORMAT ERRREC RPG NAME ERRREC
                    EXTERNAL FORMAT RSPERR RPG NAME RSPERR
     300  FCPGICF  CF  E                    WORKSTN
     400  F                                              KNUM        1
     500  F                                              KINFDS FEEDBK
     600  F                                              KINFSR FAIL
     700  F                                              KID    PGMDEV
     800  I**************************************************************
     900  I*                                                            *
    1000  I* DATA STRUCTURE FOR RPG ERROR/EXCEPTION RETURN CODES        *
    1100  I*                                                            *
    1200  I**************************************************************
           RECORD FORMAT(S):  LIBRARY FNCLIB FILE CPGICF.
                    EXTERNAL FORMAT DATAREC RPG NAME DATAREC
                    EXTERNAL FORMAT DATAEND RPG NAME DATAEND
                    EXTERNAL FORMAT INIT RPG NAME INIT
                    EXTERNAL FORMAT TERM RPG NAME TERM
 A000000   INPUT  FIELDS FOR RECORD CPGREC FILE CPGDBF FORMAT CPGREC.
 A000001                                           1  80 CPG
 B000000   INPUT  FIELDS FOR RECORD DATAREC FILE CPGICF FORMAT DATAREC.
 B000001                                           1 256 DATA
 C000000   INPUT  FIELDS FOR RECORD DATAEND FILE CPGICF FORMAT DATAEND.
 C000001                                           1 256 DATA
 D000000   INPUT  FIELDS FOR RECORD INIT FILE CPGICF FORMAT INIT.
 D000001                                           1   2 DATA2
 E000000   INPUT  FIELDS FOR RECORD TERM FILE CPGICF FORMAT TERM.
 E000001                                           1   2 DATA2
2   1300  IFEEDBK      DS
    1400  I                                       38  45 FMTNM
    1500  I                                      273 282 PGMDEV
    1600  I                                      401 402 MAJOR
    1700  I                                      403 404 MINOR
```

Figure E-9 (Part 1 of 6). RPG/400 Program for ICF Finance

```
    1800  I**************************************************************
    1900  I*                                                            *
    2000  I* WORK AREA                                                  *
    2100  I*                                                            *
    2200  I**************************************************************
3   2300  IWRKBUF      DS
    2400  I                                       1 256 WRKREC
    2500  I                                       1   1 DAT1
    2600  I                                       2   2 DAT2
SEQUENCE                                                                        IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...* USE    NUM   UPDATE     LINE   ID
4   2700  ICNSTDS      DS
    2800  I                                       1   2 INTCMD
    2900  I                                       1   1 B01
    3000  I                                       2   2 B03
    3100  I                                       3   4 TRMCMD
    3200  I                                       3   3 B05
    3300  I                                       4   4 B88
5   3400  IFILL00      DS
    3500  I                                       1 256 NULLS
    3600  I                                       1   1 BF0
    3700  I                                       1   2 BF1
    3800  I                                       1   4 BF2
    3900  I                                       1   8 BF3
    4000  I                                       1  16 BF4
    4100  I                                       1  32 BF5
    4200  I                                       1  64 BF6
    4300  I                                       1 128 BF7
    4400  I                                       2   2 BS0
    4500  I                                       3   4 BS1
    4600  I                                       5   8 BS2
    4700  I                                       9  16 BS3
    4800  I                                      17  32 BS4
    4900  I                                      33  64 BS5
    5000  I                                      65 128 BS6
    5100  I                                     129 256 BS7
    5200  C**************************************************************
    5300  C* START PROGRAM                                              *
    5400  C*                                                            *
    5500  C*   FILES ARE IMPLICITLY OPENED, AND THE SYSTEM MONITOR      *
    5600  C*   PROGRAM DEVICE IS ACQUIRED.                              *
    5700  C*                                                            *
    5800  C**************************************************************
6   5900  C           'PGMDEV'  ACQ  CPGICF
    6000  C                     MOVEL'PGMDEV'  PGMDEV 10
    6100  C**************************************************************
    6200  C*                                                            *
    6300  C* SET THE CONSTANTS NEEDED IN THE PROGRAM, AND SEND THE      *
    6400  C* INITIALIZATION SEQUENCE TO THE FINANCE CONTROLLER.         *
    6500  C*                                                            *
    6600  C**************************************************************
7   6700  C                     EXSR SETCON
    6800  C                     MOVE INTCMD    DATA2
    6900  C                     WRITEINIT
    7000  C**************************************************************
    7100  C*                                                            *
    7200  C* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATA BASE FILE*
    7300  C* 'CPGDBF'.  WHEN THE FILE REACHES END OF FILE, INDICATOR 99 *
    7400  C* IS TURNED ON. CONDITION N99 IS USED ON THE WRITE TO AVOID  *
    7500  C* SENDING A BLANK RECORD.                                    *
    7600  C*                                                            *
    7700  C**************************************************************
8   7800  C                     MOVE NULLS     WRKREC
```

Figure E-9 (Part 2 of 6). RPG/400 Program for ICF Finance

```
SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    7900  C                     READ CPGREC                   99                    3
    8000  C                     MOVELCPG       WRKREC
    8100  C           *IN99     DOUEQ'1'                                               B001   06/03/89
    8200  C                     READ CPGREC                   99                    3   001
    8300  C                     MOVELWRKREC    DATA                                     001
    8400  C                     MOVE NULLS     WRKREC                                   001
    8500  C                     MOVELCPG       WRKREC                                   001
    8600  C  N99                WRITEDATAREC                                            001
    8700  C                     END                                                    E001   06/03/89
    8800  C                     WRITEDATAEND
    8900  C**************************************************************
    9000  C*                                                            *
    9100  C* THE FOLLOWING SECTION OF CODE WRITES THE TERMINATION       *
    9200  C* CODE '0588'X TO THE CONTROLLER.                            *
    9300  C*                                                            *
    9400  C**************************************************************
    9500  C                     MOVE TRMCMD    DATA2
 9  9600  C                     WRITETERM
    9700  C**************************************************************
    9800  C*                                                            *
    9900  C* THIS LINE READS THE RESPONSE TO THE '0588'. IF THE RESPONSE*
   10000  C* IS '0581'X, THEN THE CONTROLLER ACCEPTS THE PREVIOUS       *
   10100  C* RECORDS, AND THE PROGRAM IS TERMINATED.  OTHERWISE,        *
   10200  C* PRINT OUT AN APPROPRIATE MESSAGE AND TERMINATE THE         *
   10300  C* PROGRAM.                                                   *
   10400  C*                                                            *
   10500  C**************************************************************
10 10600  C                     READ DATAREC                  98                    3
   10700  C                     MOVELDATA      WRKREC
   10800  C           DAT1      COMP HEX05                    95                    3
   10900  C         95DAT2      COMP HEX81                    95                    3
   11000  C        N95          EXSR BADRSP
   11100  C**************************************************************
   11200  C*                                                            *
   11300  C*   END THIS PROGRAM BY RELEASING THE SYSTEM MONITOR DEVICE. *
   11400  C*                                                            *
   11500  C**************************************************************
11 11600  C           PGMDEV    REL  CPGICF
   11700  C                     MOVE '1'       *INLR
   11800  C**************************************************************
   11900  C*                                                            *
   12000  C*   "SETCON" SUBROUTINE. SET CONSTANTS USED IN THIS PROGRAM. *
   12100  C*                                                            *
   12200  C**************************************************************
12 12300  CSR         SETCON    BEGSR
   12400  C**************************************************************
   12500  C*     SET UP CONSTANTS FOR "INIT" AND "TERM" SEQUENCE.       *
   12600  C**************************************************************
   12700  CSR                   BITOF'1'       B88
   12800  CSR                   BITOF'1'       B05
   12900  CSR                   BITOF'1'       B03
   13000  CSR                   BITOF'1'       B01
   13100  CSR                   BITON'04'      B88
   13200  CSR                   BITON'57'      B05
```

Figure E-9 (Part 3 of 6). RPG/400 Program for ICF Finance

```
 SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
 NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    13300  CSR                  BITON'67'      B03
    13400  CSR                  BITON'7'       B01
    13500  C*************************************************************
    13600  C*     SET UP CONSTANTS FOR FILL00.                          *
    13700  C*************************************************************
    13800  CSR                  BITOF'01234567'BF0
    13900  CSR                  MOVE BF0       BS0
    14000  CSR                  MOVE BF1       BS1
    14100  CSR                  MOVE BF2       BS2
    14200  CSR                  MOVE BF3       BS3
    14300  CSR                  MOVE BF4       BS4
    14400  CSR                  MOVE BF5       BS5
    14500  CSR                  MOVE BF6       BS6
    14600  CSR                  MOVE BF7       BS7
    14700  C*************************************************************
    14800  C*     SET UP RESPONSE CONSTANT.                             *
    14900  C*************************************************************
    15000  CSR                  BITOF'01234567'HEX00   1
    15100  CSR                  MOVE HEX00     HEX05   1
    15200  CSR                  BITON'57'      HEX05
    15300  CSR                  MOVE HEX00     HEX81   1
    15400  CSR                  BITON'07'      HEX81
    15500  CSR                  ENDSR
    15600  C*************************************************************
    15700  C*                                                           *
    15800  C*   "BADRSP" SUBROUTINE. EXCEPTION HANDLER WHICH USES        *
    15900  C*   A WRITE OPERATION TO A PRINTER FILE TO RECORD            *
    16000  C*   A RESPONSE THAT WAS NOT EXPECTED.                        *
    16200  C*                                                           *
    16300  C*************************************************************
13  16400  CSR         BADRSP    BEGSR
    16500  CSR                  WRITERSPERR
    16600  CSR                  ENDSR
    16700  C*************************************************************
    16800  C*                                                           *
    16900  C*   "FAIL" SUBROUTINE. EXCEPTION HANDLER WHICH USES          *
    17000  C*   A WRITE OPERATION TO A PRINTER FILE TO RECORD            *
    17100  C*   INFORMATION REGARDING AN ERROR CONDITION.                *
    17200  C*                                                           *
    17300  C*************************************************************
14  17400  CSR         FAIL      BEGSR
    17500  CSR                  WRITEERRREC
    17600  CSR                  ENDSR'*CANCL'
  F000000   OUTPUT FIELDS FOR RECORD ERRREC FILE CPGPRT FORMAT ERRREC.
  F000001                           PGMDEV    10  CHAR   10
  F000002                           FMTNM     18  CHAR    8
  F000003                           MAJOR     20  CHAR    2
  F000004                           MINOR     22  CHAR    2
  G000000   OUTPUT FIELDS FOR RECORD RSPERR FILE CPGPRT FORMAT RSPERR.
  H000000   OUTPUT FIELDS FOR RECORD DATAREC FILE CPGICF FORMAT DATAREC.
  H000001                           DATA     256  CHAR  256
  I000000   OUTPUT FIELDS FOR RECORD DATAEND FILE CPGICF FORMAT DATAEND.
  I000001                           DATA     256  CHAR  256
```

Figure E-9 (Part 4 of 6). RPG/400 Program for ICF Finance

```
 SEQUENCE                                                                          IND    DO    LAST       PAGE   PROGRAM
 NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7....*  USE    NUM   UPDATE     LINE   ID
  J000000   OUTPUT FIELDS FOR RECORD INIT FILE CPGICF FORMAT INIT.
  J000001                            DATA2      2  CHAR    2
  K000000   OUTPUT FIELDS FOR RECORD TERM FILE CPGICF FORMAT TERM.
  K000001                            DATA2      2  CHAR    2
           * * * * *  E N D   O F   S O U R C E   * * * * *
          A d d i t i o n a l   D i a g n o s t i c   M e s s a g e s
* 7086       100   RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE CPGDBF.
* 7089       300   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE CPGICF.
                   K e y   F i e l d   I n f o r m a t i o n
                       PHYSICAL    LOGICAL
        FILE/RCD       FIELD       FIELD       ATTRIBUTES
    01  CPGDBF
             FILE NOT KEYED DATA BASE FILE
                         C r o s s   R e f e r e n c e
 File and Record References:
        FILE/RCD    DEV/RCD     REFERENCES (D=DEFINED)
    01  CPGDBF      DISK            100D
          CPGREC                    100D A000000     7900     8200
    03  CPGICF      WORKSTN         300D    5900    11600
          DATAEND                   300D C000000     8800  I000000
          DATAREC                   300D B000000     8600    10600  H000000
          INIT                      300D D000000     6900  J000000
          TERM                      300D E000000     9600  K000000
    02  CPGPRT      PRINTER         200D
          ERRREC                    200D   17500  F000000
          RSPERR                    200D   16500  G000000
 Field References:
        FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
        *INLR       A(1)      11700M
        *IN99       A(1)       8100
        BADRSP      BEGSR     11000    16400D
        BF0         A(1)       3600D   13800M   13900
        BF1         A(2)       3700D   14000
        BF2         A(4)       3800D   14100
        BF3         A(8)       3900D   14200
        BF4         A(16)      4000D   14300
        BF5         A(32)      4100D   14400
        BF6         A(64)      4200D   14500
        BF7         A(128)     4300D   14600
        BS0         A(1)       4400D   13900M
        BS1         A(2)       4500D   14000M
        BS2         A(4)       4600D   14100M
        BS3         A(8)       4700D   14200M
        BS4         A(16)      4800D   14300M
        BS5         A(32)      4900D   14400M
        BS6         A(64)      5000D   14500M
        BS7         A(128)     5100D   14600M
        B01         A(1)       2900D   13000M   13400M
        B03         A(1)       3000D   12900M   13300M
        B05         A(1)       3200D   12800M   13200M
        B88         A(1)       3300D   12700M   13100M
* 7031  CNSTDS      DS(4)      2700D
        CPG         A(80)   A000001D    8000     8500
        DATA        A(256)  B000001D C000001D    8300M   10700  H000001D
                            I000001D
        DATA2       A(2)    D000001D E000001D    6800M    9500M J000001D
                            K000001D
        DAT1        A(1)       2500D   10800
```

Figure E-9 (Part 5 of 6). RPG/400 Program for ICF Finance

```
        DAT2        A(1)       2600D   10900
        FAIL        BEGSR       300    17400D
        FEEDBK      DS(404)     300     1300D
* 7031  FILL00      DS(256)    3400D
        FMTNM       A(8)       1400D F000002D
        HEX00       A(1)      15000D   15100    15300
        HEX05       A(1)      10800    15100D   15200M
        HEX81       A(1)      10900    15300D   15400M
        INTCMD      A(2)       2800D    6800
        MAJOR       A(2)       1600D F000003D
        MINOR       A(2)       1700D F000004D
        NULLS       A(256)     3500D    7800     8400
        PGMDEV      A(10)      1500D    6000D   11600  F000001D
        SETCON      BEGSR      6700    12300D
        TRMCMD      A(2)       3100D    9500
* 7031  WRKBUF      DS(256)    2300D
        WRKREC      A(256)     2400D    7800M    8000M    8300     8400M
                               8500M   10700M
        '*CANCL'    LITERAL   17600
        'PGMDEV'    LITERAL    5900     6000
        '01234567'  LITERAL   13800    15000
        '04'        LITERAL   13100
        '07'        LITERAL   15400
        '1'         LITERAL    8100    11700    12700    12800    12900
                              13000
        '57'        LITERAL   13200    15200
        '67'        LITERAL   13300
        '7'         LITERAL   13400
 Indicator References:
        INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
        *IN           8100
        LR           11700M
* 7031  66             200D
        95           10800M   10900    10900M   11000
* 7031  98           10600M
        99            7900M    8100     8200M    8600
      * * * * *   E N D   O F   C R O S S   R E F E R E N C E   * * * * *
                         M e s s a g e   S u m m a r y
* QRG7031 Severity:  00   Number:    5
          Message . . . . :   The Name or indicator is not referenced.
* QRG7086 Severity:  00   Number:    1
          Message . . . . :   The RPG handles blocking function for file.
            INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity:  00   Number:    1
          Message . . . . :   The RPG provides Separate-Indicator area for
            file.
      * * * * *   E N D   O F   M E S S A G E   S U M M A R Y   * * * * *
                           F i n a l   S u m m a r y
 Message Count:  (by Severity Number)
            TOTAL    00     10     20     30     40     50
              7       7      0      0      0      0      0
 Program Source Totals:
    Records . . . . . . . . . . :   176
    Specifications  . . . . . . :   96
    Table Records . . . . . . . :   0
    Comments  . . . . . . . . . :   80
 PRM has been called.
 Program CPGPGM is placed in library FNCLIB. 00 highest Error-Severity-Code.
          * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure E-9 (Part 6 of 6). RPG/400 Program for ICF Finance

ILE C/400 Source Program for Local System—Prestarted Job Example

This section describes the objects needed on the local system to run the ILE C/400 account inquiry finance program using a prestarted job example.

Configuration

Use the following configuration commands to create the synchronous data link control (SDLC) line, controller, and device descriptions used by the local system.

```
CRTLINSDLC LIND(FNCLINE) RSRCNAME(LIN022) ONLINE(*NO)
           ROLE(*PRI) NRZI(*YES)
CRTCTLFNC  CTLD(FNCCTL) TYPE(4702) MODEL(0) LINKTYPE(*SDLC)
           ONLINE(*NO) LINE(FNCLINE) STNADR(01)
CRTDEVFNC  DEVD(K002DEV) TYPE(*FNCICF) LOCADR(03)
           RMTLOCNAME(K002DEV) ONLINE(*NO) CTL(FNCCTL)
```

Use the following configuration commands to add a prestarted job to the subsystem description QCMN and start the subsystem, which automatically starts jobs for the prestarted job entry.

```
ADDPJE SBSD(QCMN)  PGM(FNCLIB/K002) USER(FNCUSER)
       CLS(FNCCLASS)
STRSBS SBSD(QCMN)
```

Note: The subsystem description QCMN is IBM-supplied and supports all communications jobs.

Program Files

The following files are used by the local system.

K002ICF
The ICF file used to send and receive records from the finance controller. This file was created by using the following command:

```
CRTICFF FILE(FNCLIB/K002ICF) SRCFILE(FNCLIB/QDDSSRC)
        SRCMBR(K002ICF)
```

The following command defines the program device entry:

```
ADDICFDEVE FILE(FNCLIB/K002ICF) PGMDEV(PGMDEV)
           RMTLOCNAME(*REQUESTER) CMNTYPE(*FINANCE)
```

An OVRICFDEVE command with the same parameters can also be used.

K002DBF
The database file that holds the account records. This file was created by using the following command:

```
CRTPF FILE(FNCLIB/K002DBF) SRCFILE(FNCLIB/QDDSSRC)
      SRCMBR(K002DBF)
```

K002PRT
The printer file used to format output to a printer. This file was created by using the following command:

```
CRTPRTF FILE(FNCLIB/K002PRT) SRCFILE(FNCLIB/QDDSSRC)
        SRCMBR(K002PRT)
```

The DDS for the ICF file used in the account withdrawal application program are shown in Figure E-10.

```
File name . . . . . . . . . . . . . . . . . . . . . :   K002ICF
  Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
File attribute  . . . . . . . . . . . . . . . . . . :   ICF
Source file containing DDS  . . . . . . . . . . . . :   QDDSSRC
  Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
Source member containing DDS  . . . . . . . . . . . :   K002ICF
Source member last changed  . . . . . . . . . . . . :   09/19/90  14:00:46
Source listing options  . . . . . . . . . . . . . . :   *SOURCE    *LIST      *NOSECLVL
DDS generation severity level . . . . . . . . . . . :   20
DDS flagging severity level . . . . . . . . . . . . :   00
Authority . . . . . . . . . . . . . . . . . . . . . :   *LIBCRTAUT
Replace file  . . . . . . . . . . . . . . . . . . . :   *YES
Text  . . . . . . . . . . . . . . . . . . . . . . . :
Compiler  . . . . . . . . . . . . . . . . . . . . . :   IBM AS/400 Data Description Processor
                                     Data Description Source
SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date
    10       A***********************************************************
    20       A*                                                         *
    30       A*                      DDS                                *
    40       A*                FOR THE ICF FILE                         *
    50       A*     USED IN ACCOUNT WITHDRAWAL APPLICATION PROGRAM      *
    60       A*                                                         *
    70       A***********************************************************
    80       A*
    90       A* FILE LEVEL INDICATORS:
   100       A*
   110       A                                      INDARA
   120       A*
   130       A***********************************************************
   140       A*                 FINANCE RECORD FORMATS                  *
   150       A***********************************************************
   160       A          R MSGBUF
   170       A                                      INVITE
   180       A            MSGREC        81A                                              09/19/90
   190       A*
   200       A          R DATABUF
   210       A                                      INVITE
   220       A            DATAREC      189A                                              09/19/90
   230       A*
   240       A          R EOSREC
   250       A                                      EOS
                        * * * * *  E N D  O F  S O U R C E  * * * * *
                                         Expanded Source
                                                                                         Field       Buffer position
SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
   110                                              INDARA
   160                  R MSGBUF                    INVITE
   180                    MSGREC        81A  B                                              81            1       1
   200                  R DATABUF                   INVITE
   220                    DATAREC      189A  B                                             189            1       1
   240                  R EOSREC                    EOS
                * * * * *  E N D  O F  E X P A N D E D  S O U R C E  * * * * *
                                         Message Summary
   Total          Informational          Warning        Error       Severe
                      (0-9)              (10-19)       (20-29)      (30-99)
       0                 0                    0             0            0
* CPC7301      00              Message . . . . :   File K002ICF created in library FNCLIB.
                        * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure E-10. DDS Source for ICF File K002ICF

The DDS for the database file used in the account withdrawal application program are shown in Figure E-11.

```
 File name . . . . . . . . . . . . . . . . . . . . . :   K002DBF
   Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
 File attribute  . . . . . . . . . . . . . . . . . . :   Physical
 Source file containing DDS  . . . . . . . . . . . . :   QDDSSRC
   Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
 Source member containing DDS  . . . . . . . . . . . :   K002DBF
 Source member last changed  . . . . . . . . . . . . :   09/20/90  16:44:37
 Source listing options  . . . . . . . . . . . . . . :   *SOURCE    *LIST      *NOSECLVL
 DDS generation severity level . . . . . . . . . . . :   20
 DDS flagging severity level . . . . . . . . . . . . :   00
 File type . . . . . . . . . . . . . . . . . . . . . :   *DATA
 Authority . . . . . . . . . . . . . . . . . . . . . :   *LIBCRTAUT
 Replace file  . . . . . . . . . . . . . . . . . . . :   *NO
 Text  . . . . . . . . . . . . . . . . . . . . . . . :
 Compiler  . . . . . . . . . . . . . . . . . . . . . :   IBM AS/400 Data Description Processor
                                      Data Description Source
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date
     10       A***********************************************************
     20       A*                                                         *
     30       A*                      DDS                                *
     40       A*             FOR THE DATABASE FILE                       *
     50       A*     USED IN ACCOUNT WITHDRAWAL APPLICATION PROGRAM      *
     60       A*                                                         *
     70       A***********************************************************
     80       A*
     90       A                                      UNIQUE
    100       A          R ACCNTINFO
    110       A            ACCTNBR       10S 0                                            09/20/90
    120       A            NAME          15
    130       A            ADDRESS       20
    140       A            CITYSTZIP     25
    150       A            OCUP          10
    160       A            BALANCE       10S 0                                            09/20/90
    170       A          K ACCTNBR
                         * * * * *  E N D  O F  S O U R C E  * * * * *
                                          Expanded Source
                                                                                          Field       Buffer position
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
     90                                              UNIQUE
    100                  R ACCNTINFO
    110                    ACCTNBR       10S 0B      COLHDG('ACCTNBR')                       10           1       1
    120                    NAME          15A  B      COLHDG('NAME')                          15          11      11
    130                    ADDRESS       20A  B      COLHDG('ADDRESS')                       20          26      26
    140                    CITYSTZIP     25A  B      COLHDG('CITYSTZIP')                     25          46      46
    150                    OCUP          10A  B      COLHDG('OCUP')                          10          71      71
    160                    BALANCE       10S 0B      COLHDG('BALANCE')                       10          81      81
    170                  K ACCTNBR
                 * * * * *  E N D  O F  E X P A N D E D  S O U R C E  * * * * *
                                          Message Summary
    Total          Informational          Warning        Error       Severe
                       (0-9)              (10-19)       (20-29)      (30-99)
        0                 0                    0             0            0
* CPC7301      00               Message . . . . :   File K002DBF created in library FNCLIB.
                         * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure E-11. DDS Source for Database File K002DBF

The DDS for the printer file used in the account withdrawal application program are shown in Figure E-12.

```
 File name . . . . . . . . . . . . . . . . . . . . . :   K002PRT
   Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
 File attribute  . . . . . . . . . . . . . . . . . . :   Printer
 Source file containing DDS  . . . . . . . . . . . . :   QDDSSRC
   Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
 Source member containing DDS  . . . . . . . . . . . :   K002PRT
 Source member last changed  . . . . . . . . . . . . :   01/15/90   9:59:01
 Source listing options  . . . . . . . . . . . . . . :   *SOURCE    *LIST      *NOSECLVL
 DDS generation severity level . . . . . . . . . . . :   20
 DDS flagging severity level . . . . . . . . . . . . :   00
 Authority . . . . . . . . . . . . . . . . . . . . . :   *LIBCRTAUT
 Replace file  . . . . . . . . . . . . . . . . . . . :   *YES
 Text  . . . . . . . . . . . . . . . . . . . . . . . :
 Compiler  . . . . . . . . . . . . . . . . . . . . . :   IBM AS/400 Data Description Processor
                                      Data Description Source
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date
     10       A****************************************************************
     20       A*                          DDS                                 *
     30       A*                  FOR THE PRINTER FILE                        *
     40       A*        USED IN ACCOUNT WITHDRAWAL APPLICATION PROGRAM        *
     50       A*                                                              *
     60       A****************************************************************
     70       A*
     80       A****************************************************************
     90       A*                   RECORD FORMATS                             *
    100       A****************************************************************
    110       A          R HEADER                    SPACEB(3)
    120       A                                    20'TRANSACTION ERROR...    '
    130       A                                      UNDERLINE
    140       A          R PRTREC                    SPACEB(3)
    150       A            PRTBUF        36A        2
                         * * * * *  E N D   O F   S O U R C E   * * * * *
                                          Expanded Source
                                                                                          Field       Buffer position
 SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
    110                  R HEADER                    SPACEB(3)
    120                                            20'TRANSACTION ERROR...    ' +            24
    130                                              UNDERLINE
    140                  R PRTREC                    SPACEB(3)
    150                    PRTBUF        36A  O     2                                        36           1
                 * * * * *  E N D   O F   E X P A N D E D   S O U R C E   * * * * *
                                          Message Summary
    Total          Informational          Warning        Error       Severe
                       (0-9)              (10-19)       (20-29)      (30-99)
        0                 0                    0             0            0
* CPC7301      00               Message . . . . :   File K002PRT created in library FNCLIB.
                         * * * * *  E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure E-12. DDS Source for Printer File K002PRT

Program Explanation: The following descriptions explain the ILE C/400 account inquiry program example in Figure E-13 on page E-32.

Note: On any type of error (for example, open error or session error), the session ends, a message prints, and the program ends.

1 This section of the program defines the structures used by the files.

2 This section defines the global constants the program uses.

3 This section defines the global variables the program uses. It also defines the common and display/ICF feedback area pointers, and the file pointers.

4 The routines, except the main routine, are prototyped so the compiler knows the type of value returned and the type of parameters passed, if any.

5 The printer file is opened for output, the ICF file is opened for record input/output, and the database file is opened for record input.

K002PRT is the printer file that sends communication information output to a printer device from the account inquiry program on the AS/400 system.

Note: The ICF file is opened with the separate indicator area option specified.

K002ICF is the ICF file that sends records to and receives records from the On-line Terminal Support (OTS) application program running on the finance controller.

K002DBF is the database file that contains the customer account information.

6 This section handles transaction requests until an unexpected error occurs or the job is ended, in which case the program also ends. Within the loop, steps 7 through 14 are run.

7 The program device (PGMDEV) used by the program is explicitly acquired. Because this is a prestarted job, the acquire is suspended until a program start request arrives.

Note: This program device was previously added to the ICF file (K002ICF) by the ADDICFDEVE command.

8 The parameters passed to the account inquiry program by the finance controller are obtained from the data area used for program initialization parameters (PIP). The account number indicator field and the account number field are the only fields retrieved from the PIP data area.

Note: The PIP data area is created for each prestarted job.

9 This section of the program checks as to whether an account number was received and whether the account number is valid (exists in the database file). If the answer to either of these checks is no, the transaction error flag is set, an appropriate error message is set in the record to be returned to the requesting device, and the error message is issued in step 10.

10 If a transaction error is detected, an error message is sent to the requesting device.

11 If no transaction error is detected, steps 12 through 13 are run.

12 The customer information and account information are copied into the record that is to be sent to the requesting device.

13 A write operation with the account inquiry response is sent to the program device that sent the inquiry request.

Note: For performance reasons, the INVITE keyword is specified as part of DATABUF record format. For more information, see “Responses” on page 7-1.

14 The ICF session is ended when the ICF record format EOSREC, which has the EOS (end-of-session) function specified, is issued.

15 This section determines whether an operation was successful by checking for a major return code of 00. If the operation was successful, a value of 0 is returned; otherwise, a value of 1 is returned.

Note: Because the feedback areas are updated after each ICF file I/O operation, the succ_rc_check function first updates the pointers to the new feedback areas before determining whether the operation was successful.

16 This section does the end-of-job processing. First, a printer record is printed with information regarding the success or failure of the program. If an I/O error occurs, the ICF session is ended by issuing an ICF record format (EOSREC), which has the EOS (end-of-session) function specified. Files used in the program are closed by passing control to step 17. Because no additional processing is needed in this program, the program returns control to the calling environment.

17 This section closes all opened files used in the program. If an error occurs on a close operation, another close is done (which is always successful).

```
Program . . . . . . . . . . . . :   K002
  Library . . . . . . . . . . . :     FNCLIB
Source file . . . . . . . . . . :   PGMSRC
  Library . . . . . . . . . . . :     FNCLIB
Source member . . . . . . . . . :   K002
Text Description  . . . . . . . :
Output  . . . . . . . . . . . . :   *PRINT
Compiler options  . . . . . . . :   *NOAGR   *NOEXPMAC   *LOGMSG   *NOSECLVL
                                :   *NOSHOWINC   *NOSHOWSKP   *NOXREF   *USRINCPATH
Checkout options  . . . . . . . :   *NOACCURACY   *NOENUM   *NOEXTERN   *NOGENERAL   *NOGOTO   *NOINIT
                                :   *NOPARM   *NOPORT   *NOPPCHECK   *NOPPTRACE
Optimization  . . . . . . . . . :   *NONE
Debugging view  . . . . . . . . :   *NONE
Define names  . . . . . . . . . :
Language level  . . . . . . . . :   *SOURCE
Source margins:
  Left margin . . . . . . . . . :   1
  Right margin  . . . . . . . . :   80
Sequence columns:
  Left Column . . . . . . . . . :
  Right Column  . . . . . . . . :
Message flagging level  . . . . :   0
Compiler messages:
  Message limit . . . . . . . . :   *NOMAX
  Message limit severity  . . . :   30
Replace module object . . . . . :   *YES
User Profile  . . . . . . . . . :   *USER
Authority . . . . . . . . . . . :   *LIBCRTAUT
Target release  . . . . . . . . :   *CURRENT
System includes . . . . . . . . :   *YES
Last change . . . . . . . . . . :   01/24/94 15:50:58
Source description  . . . . . . :
Compiler  . . . . . . . . . . . :   IBM ILE C/400 Compiler

*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.......
/*********************************************************************/
/*                                                                   */
/* This program handles account inquiry requests from a 4702        */
/* finance controller.                                               */
/*                                                                   */
/* NOTE: On any type of error (i.e open error, session errors...),   */
/* the session will be terminated, a message printed, and the       */
/* program will end.                                                 */
/*                                                                   */
/*********************************************************************/
/*********************************************************************/
/* Retrieve various structures/utilities that are used in program.   */
/*********************************************************************/
#include <stdio.h>               /* Standard I/O header.           */
#include <stdlib.h>              /* General utilities.             */
#include <stddef.h>              /* Standard definitions.          */
#include <string.h>              /* String handling utilities.     */
#include <xxdtaa.h>              /* Data area handling utilities.  */
#include <xxfdbk.h>              /* Feedback area structures.      */
#include <recio.h>               /* Record I/O routines            */
/*********************************************************************/
/* Define the structure used for the data retrieved from the data    */
/* area which contains the data sent by the finance controller.      */
/*********************************************************************/
```

Figure E-13 (Part 1 of 7). ILE C/400 Program

```
     *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
1   |struct                                                                    |
    |      {                                                                   |
    |       char   ws_type??(2??);                                             |
    |       char   ctl_unit??(2??);                                            |
    |       char   ws_num??(2??);                                              |
    |       char   audit_num??(2??);                                           |
    |       char   tellerno??(3??);                                            |
    |       char   linenbr??(2??);                                             |
    |       char   trans_code??(3??);                                          |
    |       char   splfnct??(1??);                                             |
    |       char   resvrd??(3??);                                              |
    |       char   ctl_utype??(1??);                                           |
    |       char   fld1p??(1??);       /* Account number field indicator. */   |
    |       char   fld2p??(1??);                                               |
    |       char   fld3p??(1??);                                               |
    |       char   fld4p??(1??);                                               |
    |       char   fld5p??(1??);                                               |
    |       char   fld6p??(1??);                                               |
    |       char   fld7p??(1??);                                               |
    |       char   input_fld1??(10??); /* Account number.                */    |
    |       char   input_fld2??(10??);                                         |
    |       char   input_fld3??(10??);                                         |
    |       char   input_fld4??(10??);                                         |
    |       char   input_fld5??(10??);                                         |
    |       char   input_fld6??(10??);                                         |
    |       char   input_fld7??(10??);                                         |
    |      } pip_data;                                                         |
    |/*********************************************************************/   |
    |/* Define the structures used for the data that is to be sent to     */   |
    |/* the finance controller.  These structures are used when doing     */   |
    |/* "writes" to the ICF file.                                         */   |
    |/*********************************************************************/   |
    |      /***************************************************************/   |
    |      /* Structure used to send error messages to the controller     */   |
    |      /* requesting the transaction.                                 */   |
    |      /***************************************************************/   |
    |struct                                                                    |
    |      {                                                                   |
    |       char   header1??(16??);                                            |
    |       char   newpage;                                                    |
    |       char   newline1;                                                   |
    |       char   msg_buf??(40??);                                            |
    |       char   newline2;                                                   |
    |       char   filler1??(10??);                                            |
    |       char   account_num??(10??);                                        |
    |       char   end_of_data1;                                               |
    |       char   end_of_data2;                                               |
    |       char   end_of_data3;                                               |
    |       char   end_of_data4;                                               |
    |      } msgrec       = { "000000          ",                              |
    |                         0x0C, 0x15,                                      |
    |                         "",                                              |
    |                         0x15,                                            |
    |                         "Account#: ",                                    |
    |                         "          ",                                    |
    |                         0xFF,0xFF,0xFF,0xFF };                           |
```

Figure E-13 (Part 2 of 7). ILE C/400 Program

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
      /***************************************************************/
      /* Structure used to send the account information to the       */
      /* controller requesting the transaction.  Information that    */
      /* is sent includes:  account number, customer name, address,  */
      /* occupation, and the account balance.                        */
      /***************************************************************/
struct
      {
       char   header1??(16??);
       char   newpage;
       char   newline1;
       char   filler1??(12??);
       char   account_num??(10??);
       char   newline2;
       char   filler2??(12??);
       char   name??(15??);
       char   newline3;
       char   filler3??(12??);
       char   address??(20??);
       char   newline4;
       char   filler4??(12??);
       char   city_state_zip??(25??);
       char   newline5;
       char   filler5??(12??);
       char   occupation??(10??);
       char   newline6;
       char   filler6??(12??);
       char   balance??(10??);
       char   end_of_data1;
       char   end_of_data2;
       char   end_of_data3;
       char   end_of_data4;
      } datarec        = { "000000          ",
                           0x0C,
                           0x15, "Account#  : ", "",
                           0x15, "Name      : ", "",
                           0x15, "Address   : ", "",
                           0x15, "City/State: ", "",
                           0x15, "Occupation: ", "",
                           0x15, "Balance   : ", "",
                           0xFF,0xFF,0xFF,0xFF };
/*********************************************************************/
/* Define structure used to contain the messages that can be sent    */
/* to the finance controller.                                        */
/*********************************************************************/
struct
      {
       char msg1??(40??);
       char msg2??(40??);
      } msg_records = { "Account number not received.           ",
                        "Account number not found.              " };
/*********************************************************************/
/* Define structure used to write to the print file.                 */
/*********************************************************************/
struct
      {
       char filler1??(13??);
       char major??(2??);
       char minor??(2??);
       char filler2??(19??);
      } print_rec;
```

Figure E-13 (Part 3 of 7). ILE C/400 Program

```
           *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.......
 142       |/**********************************************************************/                          |  142
 143       |/* Define the database file record structure.                         */                          |  143
 144       |/**********************************************************************/                          |  144
 145       |struct                                                                                            |  145
 146       |      {                                                                                           |  146
 147       |       char   account_num??(10??);                                                                |  147
 148       |       char   name??(15??);                                                                       |  148
 149       |       char   address??(20??);                                                                    |  149
 150       |       char   city_state_zip??(25??);                                                             |  150
 151       |       char   occupation??(10??);                                                                 |  151
 152       |       char   balance??(10??);                                                                    |  152
 153       |      } account_info_rec;                                                                         |  153
 154       |/**********************************************************************/                          |  154
 155 2     |/* Define constants/flags used in program.                             */                          |  155
 156       |/**********************************************************************/                          |  156
 157       |#define ERROR 1                   /* Error occurred.                   */                          |  157
 158       |#define NOERROR 0                                                                                 |  158
 159       |#define NORM_END 1                /* Print normal end message.         */                          |  159
 160       |#define OPEN_ERR 2                /* Print open file error message.    */                          |  160
 161       |#define ACQ_ERR 3                 /* Print acquire error message.      */                          |  161
 162       |#define IO_ERR 4                  /* Print I/O error message.          */                          |  162
 163       |#define TRUE   1                  /* Used in never-ending loop.        */                          |  163
 164       |#define FALSE  0                                                                                  |  164
 165       |/**********************************************************************/                          |  165
 166 3     |/* Declare global variables/functions.                                 */                          |  166
 167       |/**********************************************************************/                          |  167
 168       |size_t          size;           /* "size_t" is a synonym for the    */                            |  168
 169       |                                /* type of the value returned by    */                            |  169
 170       |                                /* the "sizeof" operator.           */                            |  170
 171       |_RFILE          *icffptr;       /* Pointer to ICF file.             */                            |  171
 172       |_RFILE          *prtfptr;       /* Pointer to print file.           */                            |  172
 173       |_RFILE          *dbfptr;        /* Pointer to database file.        */                            |  173
 174       |_XXIOFB_T        *comm_fdbk;    /* Pointer common I/O feedback.     */                            |  174
 175       |_XXIOFB_DSP_ICF_T *dsp_icf_fdbk; /* Pointer display/ICF I/O feedback.*/                            |  175
 176       |_RIOFB_T        *db_fdbk;       /* I/O Feedback - data base file    */                            |  176
 177 4     |int succ_rc_check(void);                                                                          |  177
 178       |void end_job(int);                                                                                |  178
 179       |void close_files(void);                                                                           |  179
 180       |/**********************************************************************/                          |  180
 181       |/* Start of mainline code.                                             */                          |  181
 182       |/**********************************************************************/                          |  182
 183       |main()                                                                                            |  183
 184       |{                                                                                                 |  184
 185       |  int          found    = FALSE; /* Found account flag.             */                            |  185
 186       |  char         dbfkey_z??(10??); /* Database file key - zoned.      */                            |  186
 187       |  char         balance_z;        /* Account balance - zoned.        */                            |  187
 188       |  int          trans_err = NOERROR; /* Transaction error indicator. */                            |  188
 189       |  _SYSindara   dsp_indic;        /* Separate indicator area.        */                            |  189
 190       |  _DTAA_NAME_T dtaname =         /* Data area name.                 */                            |  190
 191       |                    {"*PDA      ", "          " };                                                |  191
 192       |/**********************************************************************/                          |  192
 193       |/* Open printer, ICF, and database files.                              */                          |  193
 194       |/**********************************************************************/                          |  194
 195 5   1 |  if (( prtfptr = _Ropen("FNCLIB/K002PRT", "ar")) == NULL)                                         |  195
 196       |    {                                                                                             |  196
 197     2 |     printf("Open failed for printer file\n");                                                    |  197
 198     3 |     exit(ERROR);                                                                                 |  198
 199       |    }                                                                                             |  199
 200       |  if (( icffptr = _Ropen("FNCLIB/K002ICF",                                                        |  200
 201     4 |        "ar+, indicators=Y")) == NULL) end_job(OPEN_ERR);                                          |  201
 202       |  if (( dbfptr = _Ropen("FNCLIB/K002DBF", "rr riofb=n"))                                          |  202
 203     6 |             == NULL) end_job(OPEN_ERR);                                                          |  203
```

Figure E-13 (Part 4 of 7). ILE C/400 Program

```
           *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.......
 204       |/*********************************************************************/                     |   204
 205       |/* The "driver" of this program is contained in the following       */                     |   205
 206       |/* loop.  In the loop, the program:                                  */                     |   206
 207       |/*    1- acquires the requester device                              */                     |   207
 208       |/*    2- retrieves the program initiation parameters (PIP data)     */                     |   208
 209       |/*    3- processes the transaction                                  */                     |   209
 210       |/*    4- ends the session                                           */                     |   210
 211       |/*                                                                  */                     |   211
 212       |/* NOTE: The program will terminate if any error occurs.            */                     |   212
 213       |/*********************************************************************/                     |   213
 214 6  8 |  while ( TRUE )                                                                           |  214
 215       |    {                                                                                      |   215
 216       |     /*****************************************************************/                   |   216
 217       |     /* Explicitly acquire the session.                               */                   |   217
 218       |     /*****************************************************************/                   |   218
 219 7  9 |    _Racquire(icffptr, "PGMDEV    ");                                                      |  219
 220   10  |     if ( succ_rc_check() == ERROR) end_job(ACQ_ERR);                                      |   220
 221       |     /*****************************************************************/                   |   221
 222       |     /* The PIP data that is to be retrieved from the data area       */                   |   222
 223       |     /* has seven indicators which indicate which fields are          */                   |   223
 224       |     /* present:                                                      */                   |   224
 225       |     /*    If field one is present, FLD1P is "1", otherwise           */                   |   225
 226       |     /*    it is " ".                                                 */                   |   226
 227       |     /*    If field two is present, FLD2P is "2", otherwise           */                   |   227
 228       |     /*    it is " ".                                                 */                   |   228
 229       |     /*    and so on . . .                                            */                   |   229
 230       |     /* The data is loaded sequentially into the available fields     */                   |   230
 231       |     /* so if the operator enters fields 1, 3, 5, and 7, the data     */                   |   231
 232       |     /* will be stored in input fields 1, 2, 3, and 4. It is          */                   |   232
 233       |     /* necessary to test for the presence of the fields and move     */                   |   233
 234       |     /* each input field into the correct field in the program.       */                   |   234
 235       |     /* Since this program only deals with account inquiry requests,  */                   |   235
 236       |     /* if an account number is not passed this program will          */                   |   236
 237       |     /* reject the transaction request.                               */                   |   237
 238       |     /*                                                               */                   |   238
 239       |     /* The following code retrieves only those parts of the PIP      */                   |   239
 240       |     /* data that this program uses, and rejects the transaction      */                   |   240
 241       |     /* request if the account number is not among the data that is   */                   |   241
 242       |     /* passed, or if the account number is not in the database file.*/                   |   242
 243       |     /*****************************************************************/                   |   243
 244 8 12 |  QXXRTVDA(dtaname, 22, sizeof(pip_data.fld1p), pip_data.fld1p);                           |  244
 245   13  |     QXXRTVDA(dtaname, 29, sizeof(dbfkey_z), dbfkey_z);                                    |   245
 246   14  |     trans_err = NOERROR;                                                                  |   246
 247 9 15 |  if (strncmp(pip_data.fld1p, "1", 1) != 0)                                                |  247
 248       |       {                                                                                   |   248
 249   16  |        trans_err = ERROR;                                                                 |   249
 250   17  |        strncpy(msgrec.msg_buf, msg_records.msg1, 40);                                     |   250
 251   18  |        strncpy(msgrec.account_num, "N/A       ", 10);                                     |   251
 252       |       }                                                                                   |   252
 253       |     else                                                                                  |   253
 254       |       {                                                                                   |   254
 255   19  |        strncpy(account_info_rec.account_num,dbfkey_z, 10);                                |   255
 256       |        db_fdbk = _Rreadk(dbfptr, &account_info_rec,                                       |   256
 257       |                    sizeof(account_info_rec), __KEY_EQ,                                    |   257
 258       |                    &account_info_rec.account_num,                                         |   258
 259   20  |                    sizeof(account_info_rec.account_num));                                 |   259
 260   21  |        if (db_fdbk->num_bytes > 0)                                                        |   260
 261   22  |           trans_err = NOERROR;                                                            |   261
 262       |        else                                                                               |   262
 263       |          {                                                                                |   263
 264   23  |           trans_err = ERROR;                                                              |   264
 265   24  |           strncpy(msgrec.msg_buf, msg_records.msg2, 40);                                  |   265
 266   25  |           strncpy(msgrec.account_num, dbfkey_z, 10);                                      |   266
 267       |          }                                                                                |   267
 268       |       }                                                                                   |   268
```

Figure E-13 (Part 5 of 7). ILE C/400 Program

```
          *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
269          |     /****************************************************************/          |   269
270          |     /* If there is a transaction error, then send the appropriate   */          |   270
271          |     /* message and end the session; otherwise, process the         */          |   271
272          |      /* transaction and send a response to the transaction
request.  */                          |   272
273          |     /****************************************************************/          |   273
274 10 26 |     if (trans_err == ERROR)                                                    |   274
275          |        {                                                                     |   275
276    27 |         _Rformat(icffptr, "MSGBUF    ");                                      |   276
277    28 |         _Rwrite(icffptr,&msgrec, sizeof(msgrec));                                |   277
278    29 |         if ( succ_rc_check() == ERROR) end_job(IO_ERR);                           |   278
279          |        }                                                                     |   279
280 11    |     else                                                                       |   280
281          |        {                                                                     |   281
282 12 31 |        strncpy(datarec.account_num, account_info_rec.account_num,10);           |   282
283    32 |         strncpy(datarec.name, account_info_rec.name, 15);                       |   283
284    33 |         strncpy(datarec.address, account_info_rec.address, 20);                 |   284
285          |         strncpy(datarec.city_state_zip,                                     |   285
286    34 |                  account_info_rec.city_state_zip, 25);                          |   286
287    35 |         strncpy(datarec.occupation, account_info_rec.occupation, 10);           |   287
288    36 |         strncpy(datarec.balance, account_info_rec.balance, 10);                 |   288
289 13 37 |        _Rformat(icffptr, "DATABUF   ");                                         |   289
290    38 |         _Rwrite(icffptr,&datarec, sizeof(datarec));                              |   290
291    39 |         if ( succ_rc_check() == ERROR) end_job(IO_ERR);                           |   291
292          |        }                                                                     |   292
293 14 41 |     _Rformat(icffptr, "EOSREC    ");                                          |   293
294    42 |      _Rwrite(icffptr,NULL, 0);                                                  |   294
295          |     }                                                                        |   295
296          |}                                                                             |   296
297          |/********************************************************************/          |   297
298          |/*                                                                  */          |   298
299          |/*   ****************                           ****************   */          |   299
300          |/*   *                  INTERNAL FUNCTIONS                     *   */          |   300
301          |/*   ****************                           ****************   */          |   301
302          |/*                                                                  */          |   302
303          |/********************************************************************/          |   303
304          |/********************************************************************/          |   304
305          |/*              CHECK FOR SUCCESSFUL OPERATION                     */          |   305
306          |/* The following function checks whether the last operation         */          |   306
307          |/* was successful.  For an operation to be successful, the major     */          |   307
308          |/* return code must be equal to '00'.  If the operation is           */          |   308
309          |/* successful, a value of 0 is returned to the caller; otherwise,    */          |   309
310          |/* a value of 1 is returned.                                         */          |   310
311          |/********************************************************************/          |   311
312 15      |succ_rc_check()                                                                |   312
313          |{                                                                             |   313
314      1 | comm_fdbk = _Riofbk(icffptr);                                                  |   314
315          | dsp_icf_fdbk = (_XXIOFB_DSP_ICF_T *)((char *)comm_fdbk +                     |   315
316      2 |                           comm_fdbk->file_dep_fb_offset);                      |   316
317      3 | if (strncmp(dsp_icf_fdbk->major_ret_code, "00", 2) == 0)                      |   317
318      4 |   return(NOERROR);                                                           |   318
319          | else                                                                        |   319
320      5 |   return(ERROR);                                                             |   320
321          |}                                                                             |   321
```

Figure E-13 (Part 6 of 7). ILE C/400 Program

```
              *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.......
 322          |/********************************************************************/                         |    322
 323          |/*                           END JOB                                  */                         |    323
 324          |/* The following function writes message and return code (if any)    */                         |    324
 325          |/* to a printer file, ends the session with the controller,          */                         |    325
 326          |/* closes the files used by the program and returns to the caller    */                         |    326
 327          |/* of this program.                                                 */                         |    327
 328          |/********************************************************************/                         |    328
 329 16       |void end_job(int mtype)                                                                        |    329
 330          |{                                                                                              |    330
 331        1 | _Rformat(prtfptr, "HEADER    ");                                                              |    331
 332        2 | _Rwrite(prtfptr,NULL, 0);                                                                     |    332
 333        3 | _Rformat(prtfptr, "PRTREC    ");                                                              |    333
 334        4 | if (mtype != OPEN_ERR)                                                                        |    334
 335          |   {                                                                                           |    335
 336        5 |   strncpy(print_rec.major, dsp_icf_fdbk->major_ret_code, 2);                                  |    336
 337        6 |   strncpy(print_rec.minor, dsp_icf_fdbk->minor_ret_code, 2);                                  |    337
 338        7 |   strncpy(print_rec.filler1, "RETURN CODE: ", 13);                                            |    338
 339        8 |   strncpy(print_rec.filler2, "                  ", 19);                                       |    339
 340        9 |   _Rwrite(prtfptr,&print_rec, sizeof(print_rec));                                             |    340
 341          |   }                                                                                           |    341
 342       10 | if (mtype == NORM_END)                                                                        |    342
 343       11 |   _Rwrite(prtfptr,"PROGRAM COMPLETED NORMALLY          ", 36);                                |    343
 344          | else                                                                                          |    344
 345       12 |   if (mtype == OPEN_ERR)                                                                      |    345
 346       13 |     _Rwrite(prtfptr,"PROGRAM COULD NOT OPEN FILE(S)      ", 36);                              |    346
 347          |   else                                                                                        |    347
 348       14 |     if (mtype == ACQ_ERR)                                                                     |    348
 349       15 |       _Rwrite(prtfptr,"PROGRAM COULD NOT ACQUIRE DEVICE    ", 36);                            |    349
 350          |     else                                                                                      |    350
 351       16 |       _Rwrite(prtfptr,"I/O ERROR OCCURRED                  ", 36);                            |    351
 352       17 | if ((mtype != OPEN_ERR) && (mtype != ACQ_ERR))                                                |    352
 353          |   {                                                                                           |    353
 354       18 |    _Rformat(icffptr, "EOSREC    ");                                                          |    354
 355       19 |    _Rwrite(icffptr,NULL, 0);                                                                  |    355
 356          |   }                                                                                           |    356
 357       20 | close_files;                                                                                  |    357
 358       21 | if (mtype == NORM_END) exit(NOERROR);                                                         |    358
 359       23 | else exit(ERROR);                                                                             |    359
 360          |}                                                                                              |    360
 361          |/********************************************************************/                         |    361
 362          |/*                       CLOSE FILES                                  */                         |    362
 363          |/* The following function closes all files.                         */                         |    363
 364          |/********************************************************************/                         |    364
 365 17       |void close_files()                                                                             |    365
 366          |{                                                                                              |    366
 367        1 | if (icffptr != NULL)                                                                          |    367
 368        2 |   _Rclose(icffptr);                                                                           |    368
 369        3 | if (prtfptr != NULL)                                                                          |    369
 370        4 |   _Rclose(prtfptr);                                                                           |    370
 371        5 | if (dbfptr != NULL)                                                                           |    371
 372        6 |   _Rclose(dbfptr);                                                                            |    372
 373          |}                                                                                              |    373
```

Figure E-13 (Part 7 of 7). ILE C/400 Program

Appendix F. Non-Intersystem Communications Function Finance Example Programs

This appendix provides COBOL/400 and RPG/400 example programs to demonstrate how finance communications is used by using the submit Finance Job (SBMFNCJOB) command.

Figure F-1 shows representations of example programs.

COBOL/400 and RPG/400 Source Programs for the Local System

This section describes the objects needed on the local system to run the ILE COBOL/400 and ILE RPG/400 account inquiry and withdrawal finance example programs.

Configuration

The following configuration commands are used to create the synchronous data link control (SDLC) line, controller, and device descriptions used by the local system.

```
CRTLINSDLC LIND(FNCLINE) RSRCNAME(LIN022) ONLINE(*NO)
           ROLE(*PRI) NRZI(*YES)
CRTCTLFNC  CTLD(FNCCTL) TYPE(4702) MODEL(0) LINKTYPE(*SDLC)
           ONLINE(*NO) LINE(FNCLINE) STNADR(01)
CRTDEVFNC  DEVD(FNCDEV) TYPE(4704) LOCADR(03) ONLINE(*NO)
           CTL(FNCCTL)
```

Program File

The following files are used by the local system:

ACCOUNT
The database file that holds the account records.

This file was created by using the following command:

```
CRTPF FILE(FNCLIB/ACCOUNT) SRCFILE(FNCLIB/QDDSSRC)
      SRCMBR(ACCOUNT)
```

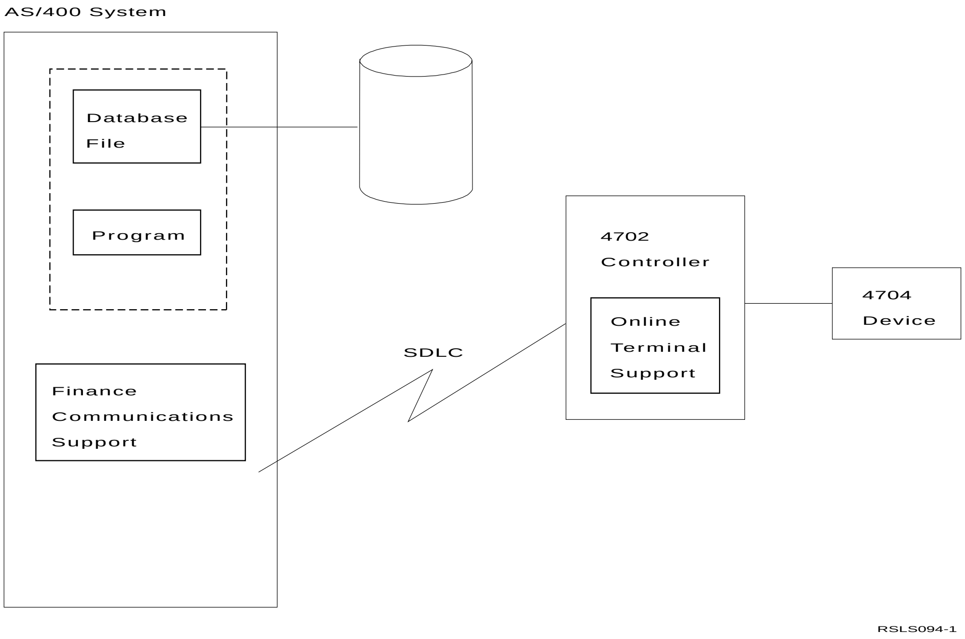


Figure F-1. Configuration Example for Non-ICF Finance

The DDS used by this file is shown in Figure F-2.

```
File name . . . . . . . . . . . . . . . . . . . . . :   ACCOUNT
  Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
File attribute  . . . . . . . . . . . . . . . . . . :   Physical
Source file containing DDS  . . . . . . . . . . . . :   QDDSSRC
  Library name  . . . . . . . . . . . . . . . . . . :   FNCLIB
Source member containing DDS  . . . . . . . . . . . :   ACCOUNT
Source member last changed  . . . . . . . . . . . . :   05/31/89  11:05:08
Source listing options  . . . . . . . . . . . . . . :   *SOURCE    *LIST      *NOSECLVL
DDS generation severity level . . . . . . . . . . . :   20
File type . . . . . . . . . . . . . . . . . . . . . :   *DATA
Authority . . . . . . . . . . . . . . . . . . . . . :   *CHANGE
Text  . . . . . . . . . . . . . . . . . . . . . . . :
Compiler  . . . . . . . . . . . . . . . . . . . . . :   IBM AS/400 Data Description Processor
                                     Data Description Source
SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  Date
    10       A************************************************************
    20       A*                                                          *
    30       A*                    DDS                                   *
    40       A*             FOR THE DATABASE FILE                        *
    50       A*     USED IN ACCOUNT INQUIRY APPLICATION PROGRAM          *
    60       A*                                                          *
    70       A************************************************************
    80       A*
    90       A                                       UNIQUE
   100       A          R ACCOUNTR
   110       A            ACCTNR         8  0
   120       A            NAME          21
   130       A            STR1           3
   140       A            STR2          18
   150       A            CITY          16
   160       A            ZIP            5
   170       A            OCUP          21
   180       A            TBAL          10  2
   190       A            DLYWTH        10  2
   200       A            DLYDEP        10  2
   210       A            LIMIT1        10  2
   220       A            LIMIT2        10  2
   230       A            RSVCS1        10  2
   240       A            RSVCS2        10  2
   250       A            RSVRM1        10  2
   260       A            RSVRM2        10  2
   270       A            ACTIVE         1  0
   280       A            LCKDSC        38
   290       A            WTHDRL        10  2
   300       A          K ACCTNR
                      * * * * *   E N D   O F   S O U R C E   * * * * *
```

Figure F-2 (Part 1 of 2). DDS Source for ACCOUNT File

```
                                         Expanded Source
                                                                                     Field       Buffer position
SEQNBR  *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 length        Out     In
    90                                       UNIQUE
   100                  R ACCOUNTR
   110                    ACCTNR         8P 0B      COLHDG('ACCTNR')                       5           1       1
   120                    NAME          21A  B      COLHDG('NAME')                        21           6       6
   130                    STR1           3A  B      COLHDG('STR1')                         3          27      27
   140                    STR2          18A  B      COLHDG('STR2')                        18          30      30
   150                    CITY          16A  B      COLHDG('CITY')                        16          48      48
   160                    ZIP            5A  B      COLHDG('ZIP')                          5          64      64
   170                    OCUP          21A  B      COLHDG('OCUP')                        21          69      69
   180                    TBAL          10P 2B      COLHDG('TBAL')                         6          90      90
   190                    DLYWTH        10P 2B      COLHDG('DLYWTH')                       6          96      96
   200                    DLYDEP        10P 2B      COLHDG('DLYDEP')                       6         102     102
   210                    LIMIT1        10P 2B      COLHDG('LIMIT1')                       6         108     108
   220                    LIMIT2        10P 2B      COLHDG('LIMIT2')                       6         114     114
   230                    RSVCS1        10P 2B      COLHDG('RSVCS1')                       6         120     120
   240                    RSVCS2        10P 2B      COLHDG('RSVCS2')                       6         126     126
   250                    RSVRM1        10P 2B      COLHDG('RSVRM1')                       6         132     132
   260                    RSVRM2        10P 2B      COLHDG('RSVRM2')                       6         138     138
   270                    ACTIVE         1P 0B      COLHDG('ACTIVE')                       1         144     144
   280                    LCKDSC        38A  B      COLHDG('LCKDSC')                      38         145     145
   290                    WTHDRL        10P 2B      COLHDG('WTHDRL')                       6         183     183
   300                  K ACCTNR
              * * * * *   E N D   O F   E X P A N D E D   S O U R C E   * * * * *
                                            Messages
   ID      Severity  Number
                                         Message Summary
   Total          Informational          Warning        Error       Severe
                      (0-9)              (10-19)       (20-29)      (30-99)
      0                 0                     0            0           0
* CPC7301      00              Message . . . . :   File ACCOUNT created in library FNCLIB.
                      * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure F-2 (Part 2 of 2). DDS Source for ACCOUNT File

COBOL/400 Program Explanation

This section explains the COBOL/400 account inquiry and withdrawal program example in Figure F-3 on page F-4.

1 This section of the program defines the database file (ACCOUNT).

ACCOUNT is the database file that contains the customer account information.

2 The program opens the file to be used.

3 The parameters passed to the account inquiry program by the finance controller are placed in a data structure. The account number and the deposit amount fields are the only fields checked in this example. If the account number is not passed by the online terminal support (OTS) program, a message is sent to the finance controller and the program is ended. If a withdrawal amount is passed by OTS, the request is a withdrawal request; otherwise, it is simply an account inquiry request.

4 A read operation from the database file (with the account number as the key) is done.

5 If the account number is not in the database file, a message is sent to the finance controller. If the account number is in the database file, control is passed to either **6** if the OTS request is an account inquiry, or to **7** if the OTS request is a withdrawal transaction.

6 This routine sends out the account inquiry reply in two records. The first record contains customer information, and the second record contains account information.

Note: Because the QFN-write (QFNWRT) routine sends the records to the finance controller, after the records have been sent, the length of data being sent is set to zero. This prevents the finance job from sending any additional data for this transaction to the finance controller.

7 This routine handles withdrawal transactions. If the account is not active, a message is sent to the finance controller and the program is ended. Otherwise, control is passed to **8** (to process the withdrawal transaction).

8 This routine determines whether there are sufficient funds to satisfy the withdrawal request. If not, the withdrawal request is handled as an inquiry request, and a message stating that sufficient funds are not available is included with the inquiry request response. If sufficient funds exist to satisfy the withdrawal request, control is passed to **9**.

9 This routine updates the database file to reflect the withdrawal request, and sends the updated account information to the finance controller by using the QFNWRT routine.

10 This routine is called to build and send the error message to the finance controller.

Note: The routine QFNWRT is not used to send the error message. Because the send depth is not zero, the finance job sends the error message automatically to the finance controller when the program ends.

11 This routine is called to build and send the customer information to the finance controller.

12 This routine is called to build and send the account information to the finance controller.

13 This routine is called to build and send the account information to the finance controller when the transaction is a withdrawal request.

14 The database file is closed and the program is ended.

```
Program  . . . . . . . . . . . . . . :   OTSCBL1
  Library  . . . . . . . . . . . . . :     FNCLIB
Source file  . . . . . . . . . . . . :   PGMSRC
  Library  . . . . . . . . . . . . . :     FNCLIB
Source member  . . . . . . . . . . . :   OTSCBL1     09/18/90 15:02:12
Generation severity level  . . . . . :   29
Text 'description' . . . . . . . . . :   *BLANK
Source listing options . . . . . . . :   *NONE
Generation options . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . :   29
Print file . . . . . . . . . . . . . :   QSYSPRT
  Library  . . . . . . . . . . . . . :     *LIBL
FIPS flagging  . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . :   *NOFLAG
Flagging severity  . . . . . . . . . :   0
Replace program  . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . :   *USER
Authority  . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
    1  000010 PROCESS  APOST MAP
    2  000020 IDENTIFICATION DIVISION.
       000030
    3  000040 PROGRAM-ID.  OTSCBL1.
       000050
       000060*************************************************************************
       000070*                                                                       *
       000080*       PROGRAM TO HANDLE ACCOUNT INQUIRY AND WITHDRAWAL                *
       000090*                   AS/400 FINANCE SUPPORT                              *
       000100*                                                                       *
       000110*************************************************************************
       000120
    4  000130 ENVIRONMENT DIVISION.
       000140
    5  000150 CONFIGURATION SECTION.
    6  000160 SOURCE-COMPUTER.  IBM-AS400.
    7  000170 OBJECT-COMPUTER.  IBM-AS400.
       000180
    8  000190 INPUT-OUTPUT SECTION.
    9  000200 FILE-CONTROL.
       000210
1  10  000220     SELECT ACCOUNT
   11  000230          ASSIGN TO DATABASE-ACCOUNT
   12  000240          ORGANIZATION IS INDEXED
   13  000250          ACCESS MODE IS RANDOM
   14  000260          RECORD KEY IS EXTERNALLY-DESCRIBED-KEY.
   15  000270 DATA DIVISION.
       000280
   16  000290 FILE SECTION.
       000300
   17  000310 FD  ACCOUNT
   18  000320     LABEL RECORDS ARE STANDARD.
   19  000330 01  ACCOUNT-REC.
   20  000340     COPY DDS-ACCOUNTR-I-O OF ACCOUNT.
      +000001*    I-O FORMAT:ACCOUNTR   FROM FILE ACCOUNT    OF LIBRARY FNCLIB             ACCOUNTR
      +000002*                                                                             ACCOUNTR
      +000003*THE KEY DEFINITIONS FOR RECORD FORMAT  ACCOUNTR                              ACCOUNTR
      +000004*  NUMBER               NAME              RETRIEVAL     TYPE    ALTSEQ        ACCOUNTR
      +000005*   0001   ACCTNR                          ASCENDING   SIGNED     NO          ACCOUNTR
   21 +000006       05  ACCOUNTR.                                                          ACCOUNTR
   22 +000007           06 ACCTNR               PIC S9(8)       COMP-3.                   ACCOUNTR
   23 +000008           06 NAME                 PIC X(21).                                ACCOUNTR
   24 +000009           06 STR1                 PIC X(3).                                 ACCOUNTR
```

Figure F-3 (Part 1 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
  25 +000010           06 STR2                   PIC X(18).                              ACCOUNTR
  26 +000011           06 CITY                   PIC X(16).                              ACCOUNTR
  27 +000012           06 ZIP                    PIC X(5).                               ACCOUNTR
  28 +000013           06 OCUP                   PIC X(21).                              ACCOUNTR
  29 +000014           06 TBAL                   PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
  30 +000015           06 DLYWTH                 PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
  31 +000016           06 DLYDEP                 PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
  32 +000017           06 LIMIT1                 PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
  33 +000018           06 LIMIT2                 PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
  34 +000019           06 RSVCS1                 PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
  35 +000020           06 RSVCS2                 PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
  36 +000021           06 RSVRM1                 PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  37 +000022           06 RSVRM2                 PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
  38 +000023           06 ACTIVE                 PIC S9(1)        COMP-3.                ACCOUNTR
  39 +000024           06 LCKDSC                 PIC X(38).                              ACCOUNTR
  40 +000025           06 WTHDRL                 PIC S9(8)V9(2)   COMP-3.                ACCOUNTR
  41  000350 WORKING-STORAGE SECTION.
      000360
  42  000370 01  FILLER.
  43  000380     05    FFFFFF PIC 9(8) COMP-4 VALUE 16777215.
  44  000390     05    FILLER REDEFINES FFFFFF.
  45  000400          10  FILLER PIC X.
  46  000410          10  FFS    PIC XXX.
      000420***************************************************************************
      000430*   FORMAT OF THE DATA SENT BY OTS TO THE AS/400                          *
      000440***************************************************************************
      000450
  47  000460 01       OTS-INPUT-DATA.
  48  000470     10      OTS-CTL-ITEMS.
  49  000480          15       WSTYPE          PIC XX.
  50  000490          15       CTLUNIT         PIC XX.
  51  000500          15       WSNO            PIC XX.
  52  000510          15       AUDTNO          PIC XX.
  53  000520          15       TELLERNO        PIC 999.
  54  000530          15       LINENBR         PIC 99.
  55  000540          15       TRNCDE          PIC XXX.
  56  000550          15       SPLFNCT         PIC X.
  57  000560          15       RESVRD          PIC XXX.
  58  000570          15       CTLUTYPE        PIC X.
  59  000580     10     OTS-DATA-AREA.
  60  000590          15       FLD1P           PIC X.
  61  000600          15       FLD2P           PIC X.
  62  000610          15       FLD3P           PIC X.
  63  000620          15       FLD4P           PIC X.
  64  000630          15       FLD5P           PIC X.
  65  000640          15       FLD6P           PIC X.
  66  000650          15       FLD7P           PIC X.
  67  000660          15       INPUT-FIELD     PIC X(10) OCCURS 7 TIMES
  68  000670                                             INDEXED BY FLD.
      000680***************************************************************************
      000690*   FORMAT OF THE DATA SENT BY THE AS/400 TO OTS                          *
      000700***************************************************************************
  69  000710 01  OTS-OUTPUT-DATA.
  70  000720     05     OTS-CONTROL.
  71  000730          10       OTS-MODE        PIC 9     VALUE IS 0.
  72  000740          10       OTS-RSV         PIC XXXX  VALUE IS '    '.
  73  000750          10       MOREDT          PIC 9     VALUE IS 0.
  74  000760          10       RSPDS           PIC X     VALUE IS ' '.
  75  000770          10       DRSPF           PIC X     VALUE IS ' '.
  76  000780          10       RSPMS           PIC X     VALUE IS ' '.
  77  000790          10       MRSPF           PIC X     VALUE IS ' '.
  78  000800          10       RSPJP           PIC X     VALUE IS ' '.
  79  000810          10       JPRSPF          PIC X     VALUE IS ' '.
  80  000820          10       JPPARM          PIC X     VALUE IS ' '.
  81  000830          10       RSPPS           PIC X     VALUE IS ' '.
  82  000840          10       PPRSPF          PIC X     VALUE IS ' '.
  83  000850          10       PSPARM          PIC X     VALUE IS ' '.
```

Figure F-3 (Part 2 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  84  000860     05     OTS-FORMATTED-DATA     PIC X(240).
      000870******************************************************************
      000880*   DATA FORMAT FOR ERROR MESSAGE DISPLAY.                       *
      000890*        ERRORS HANDLED ARE . . . NO ACCOUNT ENTERED OR          *
      000900*                                 INVALID ACCOUNT ENTERED        *
      000910******************************************************************
  85  000920 01  FORMATTED-DATA-1.
  86  000930     05   NEW-PAGE-1     PIC X.
  87  000940     05   TEXT-1         PIC X(30).
  88  000950     05   ACCTNR         PIC ZZZZZZZZ.
  89  000960     05   DATA-SEP-1     PIC X.
  90  000970     05   LASTFF-1       PIC XXX.
      000980
      000990******************************************************************
      001000*   DATA FORMAT FOR FIRST HALF OF VALID INQUIRY DISPLAY.        *
      001010******************************************************************
      001020
  91  001030 01  FORMATTED-DATA-2.
  92  001040     05   NEW-PAGE-2     PIC X.
  93  001050     05   SETPOS-2A      PIC X.
  94  001060     05   MOVHOR-2A      PIC X.
  95  001070     05   HEX20-2A       PIC X.
  96  001080     05   DATE-2         PIC ZZ/ZZ/ZZ.
  97  001090     05   NEWLIN-2B      PIC X.
  98  001100     05   FILLER         PIC X(5) VALUE 'ACC#:'.
  99  001110     05   ACCTNR         PIC ZZZZZZZ9.
 100  001120     05   FILLER         PIC X.
 101  001130     05   NAME           PIC X(21).
 102  001140     05   NEWLIN-2C      PIC X.
 103  001150     05   SETPOS-2C      PIC X.
 104  001160     05   MOVHOR-2C      PIC X.
 105  001170     05   HEX0F-2C       PIC X.
 106  001180     05   STR1           PIC XXX.
 107  001190     05   STR2           PIC X(18).
 108  001200     05   NEWLIN-2D      PIC X.
 109  001210     05   SETPOS-2D      PIC X.
 110  001220     05   MOVHOR-2D      PIC X.
 111  001230     05   HEX0F-2D       PIC X.
 112  001240     05   CITY           PIC X(16).
 113  001250     05   NEWLIN-2E      PIC X.
 114  001260     05   SETPOS-2E      PIC X.
 115  001270     05   MOVHOR-2E      PIC X.
 116  001280     05   HEX0F-2E       PIC X.
 117  001290     05   OCUP           PIC X(21).
 118  001300     05   DATA-SEP-2     PIC X.
 119  001310     05   LASTFF-2       PIC XXX.
      001320******************************************************************
      001330*   DATA FORMAT FOR SECOND HALF OF VALID INQUIRY DISPLAY.       *
      001340******************************************************************
      001350
 120  001360 01  FORMATTED-DATA-3.
 121  001370     05   SETPOS-3       PIC X.
 122  001380     05   SETLIN-3       PIC X.
 123  001390     05   HEX06-3        PIC X.
 124  001400     05   FILLER         PIC X(5) VALUE 'BAL :'.
```

Figure F-3 (Part 3 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
 125  001410     05   TBAL           PIC ZZ,ZZZ,ZZZ.99-.
 126  001420     05   FILLER         PIC X(5).
 127  001430     05   FILLER         PIC X(10) VALUE 'TOT.RSRVD:'.
 128  001440     05   NEWLIN-3A      PIC X.
 129  001450     05   FILLER         PIC X(5) VALUE 'DEP :'.
 130  001460     05   DLYDEP         PIC ZZ,ZZZ,ZZZ.99.
 131  001470     05   FILLER         PIC X(3).
 132  001480     05   TOTRSV-3       PIC ZZ,ZZZ,ZZZ.99-.
 133  001490     05   NEWLIN-3B      PIC X.
 134  001500     05   FILLER         PIC X(5) VALUE 'WTH :'.
 135  001510     05   DLYWTH         PIC ZZ,ZZZ,ZZZ.99.
 136  001520     05   NEWLIN-3C      PIC X.
 137  001530     05   FILLER         PIC X(5) VALUE 'LMT1:'.
 138  001540     05   LIMIT1         PIC ZZ,ZZZ,ZZZ.99.
 139  001550     05   NEWLIN-3D      PIC X.
 140  001560     05   FILLER         PIC X(5) VALUE 'LMT2:'.
 141  001570     05   LIMIT2         PIC ZZ,ZZZ,ZZZ.99.
 142  001580     05   NEWLIN-3E      PIC X.
 143  001590     05   FILLER         PIC X(5) VALUE 'W/B :'.
 144  001600     05   WRKBAL-3       PIC ZZ,ZZZ,ZZZ.99.
 145  001610     05   NEWLIN-3F      PIC X.
 146  001620     05   LCKDSC         PIC X(38).
 147  001630     05   DATA-SEP-3     PIC X.
 148  001640     05   LASTFF-3       PIC XXX.
      001650*************************************************************************
      001660*   DATA FORMAT FOR VALID WITHDRAWAL DISPLAY                            *
      001670*************************************************************************
 149  001680 01  FORMATTED-DATA-4.
 150  001690     05   NEWPAG-4       PIC X.
 151  001700     05   SETPOS-4A      PIC X.
 152  001710     05   MOVHOR-4A      PIC X.
 153  001720     05   HEX20-4A       PIC X.
 154  001730     05   DATE-4         PIC ZZ/ZZ/ZZ.
 155  001740     05   NEWLIN-4B      PIC X.
 156  001750     05   FILLER         PIC X(5) VALUE 'ACC#:'.
 157  001760     05   ACCTNR         PIC ZZZZZZZ9.
 158  001770     05   FILLER         PIC X.
 159  001780     05   NAME           PIC X(21).
 160  001790     05   NEWLIN-4C      PIC X.
 161  001800     05   SETPOS-4C      PIC X.
 162  001810     05   MOVHOR-4C      PIC X.
 163  001820     05   HEX0F-4C       PIC X.
 164  001830     05   OCUP           PIC X(21).
 165  001840     05   NEWLIN-4D      PIC X.
 166  001850     05   FILLER         PIC X(10) VALUE 'AMOUNT :'.
 167  001860     05   AMOUNT-4       PIC ZZ,ZZZ,ZZZ.99.
 168  001870     05   NEWLIN-4E      PIC X.
 169  001880     05   NEWLIN-4F      PIC X.
 170  001890     05   FILLER         PIC X(10) VALUE 'BALANCE : '.
 171  001900     05   BALOUT-4       PIC ZZ,ZZZ,ZZZ.99-.
 172  001910     05   NEWLIN-4G      PIC X.
 173  001920     05   FILLER         PIC X(10) VALUE 'WORK LIMIT'.
 174  001930     05   WRKBAL-4       PIC ZZ,ZZZ,ZZZ,99-.
 175  001940     05   NEWLIN-4H      PIC X.
 176  001950     05   NEWLIN-4I      PIC X.
```

Figure F-3 (Part 4 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
 177  001960     05   TEXT-4         PIC X(20).
 178  001970     05   DATA-SEP-4A    PIC X.
 179  001980     05   LASTFF-4       PIC XXX.
      001990*************************************************************************
      002000*   DATA AREA TO DEFINE THE HEX VALUES NEEDED IN THE                    *
      002010*   PROGRAM TO SEND CONTROL CHARACTERS TO THE 4700                      *
      002020*************************************************************************
      002030
 180  002040 01  DUMMY1.
      002050*    05   HEX0C          PIC X.
      002060*    05   HEX15          PIC X.
 181  002070     06   BIN1           PIC 9999 COMP-4 VALUE IS 3093.
      002080*    05   HEX08          PIC X.
      002090*    05   HEX34          PIC X.
 182  002100     06   BIN2           PIC 9999 COMP-4 VALUE IS 2100.
      002110*    05   HEX04          PIC X.
      002120*    05   HEXFF          PIC X.
 183  002130     06   BIN3           PIC 9999 COMP-4 VALUE IS 1279.
      002140*    05   HEX02          PIC X.
      002150*    05   HEX12          PIC X.
 184  002160     06   BIN4           PIC 9999 COMP-4 VALUE IS 530.
      002170*    05   HEX0D          PIC X.
      002180*    05   HEX25          PIC X.
 185  002190     06   BIN5           PIC 9999 COMP-4 VALUE IS 3365.
      002200*    05   HEX0F          PIC X.
      002210*    05   HEX20          PIC X.
 186  002220     06   BIN6           PIC 9999 COMP-4 VALUE IS 3872.
      002230*    05   HEX0A          PIC X.
      002240*    05   HEX06          PIC X.
 187  002250     06   BIN7           PIC 9999 COMP-4 VALUE IS 2566.
      002260*    05   DUMMY00        PIC X.
      002270*    05   LINENBR-HEX    PIC X.
 188  002280     06   BIN-LINENBR    PIC 9999 COMP-4.
      002290*************************************************************************
      002300*   REDEFINES AREA TO GIVE THE HEX CONTROL CODES MEANINGFUL NAMES       *
      002310*************************************************************************
      002320
 189  002330 01  DUMMY2                             REDEFINES DUMMY1.
 190  002340     05   NEWPAG         PIC X.
 191  002350     05   NEWLIN         PIC X.
 192  002360     05   MOVHOR         PIC X.
 193  002370     05   SETPOS         PIC X.
 194  002380     05   SETLIN         PIC X.
 195  002390     05   DTASEP         PIC X.
 196  002400     05   HEX02          PIC X.
 197  002410     05   HEX12          PIC X.
 198  002420     05   HEX0D          PIC X.
 199  002430     05   HEX25          PIC X.
 200  002440     05   HEX0F          PIC X.
 201  002450     05   HEX20          PIC X.
 202  002460     05   HEX0A          PIC X.
 203  002470     05   HEX06          PIC X.
 204  002480     05   DUMMY00        PIC X.
 205  002490     05   LINENBR-HEX    PIC X.
      002500*************************************************************************
```

Figure F-3 (Part 5 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      002510*   THIS IS THE 4700 CONTROL BYTE PASSED AS A PARAMETER TO             *
      002520*   QFNWRT.  IT IS SET TO INDICATE THAT TRANSACTION DATA FOLLOWS.      *
      002530*************************************************************************
      002540
  206 002550 01  CONTROL-BYTE.
  207 002560     05   DATA-TYPE      PIC X     VALUE '0'.
      002570*************************************************************************
      002580*   THESE ARE THE MESSAGES USED BY THE PROGRAM                         *
      002590*************************************************************************
      002600
  208 002610 01  MESSAGES.
  209 002620   05   MSG1      PIC X(30) VALUE 'NO ACCOUNT NR. ENTERED'.
  210 002630   05   MSG2      PIC X(30) VALUE 'ACCOUNT NR. NOT FOUND'.
  211 002640   05   MSG3      PIC X(30) VALUE 'ACCOUNT NO LONGER ACTIVE'.
  212 002650   05   MSG4      PIC X(30) VALUE 'INSUFFICIENT FUNDS AVAILABLE'.
      002660
  213 002670 01  MSG-TABLE                         REDEFINES MESSAGES.
  214 002680     05   MSG       PIC X(30) OCCURS 4 TIMES.
      002690
  215 002700 01  ACCOUNT-STATUS PIC X.
  216 002710     88   VALID-ACCOUNT VALUE IS '1'.
  217 002720     88   INVALID-ACCOUNT VALUE IS ' '.
      002730
      002740
  218 002750 01  EDATE          PIC  999999.
  219 002760 01  ACCTNO         PIC  S9(8).
  220 002770 01  AMOUNT         PIC  9(11)V99 COMP.
  221 002780 01  AMOUNT-IN      PIC  9(10).
  222 002790 01  WRKBAL         PIC  9(8)V99  COMP-3.
  223 002800 01  TOTRSV         PIC  9(8)V99  COMP-3.
  224 002810 01  TOTAVL         PIC  9(8)V99  COMP-3.
      002820*************************************************************************
      002830*   THESE ARE THE PARAMETERS PASSED TO THE PROGRAM BY THE              *
      002840*   FINANCE SUPPORT FACILITY.                                          *
      002850*************************************************************************
  225 002860 LINKAGE SECTION.
  226 002870 01  WSID           PIC X(10).
  227 002880 01  SNDLEN         PIC S9(10)V9(5) USAGE IS COMP.
  228 002890 01  DATA-PARM      PIC X(256).
  229 002900 PROCEDURE DIVISION USING WSID, SNDLEN, DATA-PARM.
      002910
      002920 INITIALIZE-PROGRAM.
2 230 002930     OPEN I-O ACCOUNT.
  231 002940     ACCEPT EDATE FROM DATE.
      002950
      002960*************************************************************************
      002970*                                                                      *
      002980* THE OTS DATA COMES INTO THE PROGRAM WITH UP TO SEVEN FIELDS.         *
      002990* OTS PROVIDES SEVEN FLAGS INDICATING WHETHER THE FIELDS ARE PRESENT.  *
      003000*  IF FIELD ONE IS PRESENT FLD1P IS "1", OTHERWISE IT IS " "           *
      003010*     TWO IS PRESENT FLD2P IS "2", OTHERWISE IT IS " "                 *
      003020*           AND SO ON . . .                                            *
      003030*  THE DATA IS LOADED SEQUENTIALLY INTO THE AVAILABLE FIELDS           *
      003040*  SO IF THE OPERATOR ENTERS FIELDS 1, 3, 5, AND 7                     *
      003050*  THE DATA WILL BE STORED IN INPUT FIELDS 1, 2, 3, AND 4.             *
```

Figure F-3 (Part 6 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
  STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
       003060*  IT IS NECESSARY TO TEST FOR THE PRESENCE OF THE FIELDS AND MOVE      *
       003070*  EACH INPUT FIELD INTO THE CORRECT FIELD IN THE PROGRAM.              *
       003080*                                                                       *
       003090*                                                                       *
       003100* TO PROCESS THE TRANSACTION - FIRST CHECK THE ACCOUNT NUMBER           *
       003110* TO SEE WHETHER IT WAS ENTERED.  IF NO ACCOUNT NUMBER WAS ENTERED,     *
       003120* SEND AN ERROR MESSAGE BACK TO THE OPERATOR.                           *
       003130*                                                                       *
       003140*************************************************************************
       003150
   232  003160     MOVE DATA-PARM TO OTS-INPUT-DATA.
   233  003170     SET FLD TO 1.
3      003180
   234  003190     IF FLD1P = '1'
       003200     THEN
   235  003210       MOVE INPUT-FIELD(FLD) TO ACCTNO
   236  003220       SET FLD TO 2                                                     09/18/90
       003230     ELSE
   237  003240        MOVE MSG(1) TO TEXT-1
   238  003250        PERFORM BUILD-FORMAT-1
   239  003260        PERFORM CLEAN-UP.
       003270
   240  003280     IF FLD2P = '2'
       003290     THEN
   241  003300       MOVE INPUT-FIELD(FLD) TO AMOUNT-IN
   242  003310       MOVE AMOUNT-IN TO AMOUNT
   243  003320       COMPUTE AMOUNT = AMOUNT / 100
       003340     ELSE
   244  003350       MOVE ZEROS TO AMOUNT.
       003360
   245  003370     PERFORM PROCESS-TRANSACTION.
       003380
       003390*************************************************************************
       003400* BEGIN PROCESSING THE TRANSACTION -                                    *
       003410* A READ FROM THE DATABASE FILE IS DONE USING THE ACCOUNT               *
       003420* NUMBER AS THE KEY.  IF THE READ WAS SUCCESSFUL, PERFORM               *
       003430* AN ACCOUNT INQUIRY OR A WITHDRAWAL TRANSACTION (DEPENDENT             *
       003440* ON WHETHER A WITHDRAWAL AMOUNT WAS PASSED); OTHERWISE, SEND           *
       003450* A MESSAGE TO THE FINANCE CONTROLLER STATING THAT THE ACCOUNT          *
       003460* NUMBER WAS NOT FOUND.                                                 *
       003470*************************************************************************
       003480 PROCESS-TRANSACTION.
   246  003490     SET VALID-ACCOUNT TO TRUE.
   247  003500     MOVE ACCTNO TO ACCTNR OF ACCOUNT-REC.
4 248  003510     READ ACCOUNT INVALID KEY SET INVALID-ACCOUNT TO TRUE.
       003520
5 250  003530     IF VALID-ACCOUNT
       003540     THEN
   251  003550       IF AMOUNT > 0
       003560       THEN
   252  003570         PERFORM ATTEMPT-WITHDRAWAL
       003580       ELSE
   253  003590         PERFORM VALID-INQUIRY
       003600     ELSE
   254  003610       MOVE MSG(2) TO TEXT-1
```

Figure F-3 (Part 7 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  255  003620       PERFORM BUILD-FORMAT-1.
       003630
       003640***************************************************************************
       003650*                                                                         *
       003660* ONCE THE PROCESSING IS COMPLETE, AND THE PROGRAM IS ENDED, THE FINANCE*
       003670* JOB WILL SEND THE DATA IN DATA-PARM BACK TO THE REQUESTING WORK       *
       003680* STATION IF SNDLEN IS GREATER THAN 0.  IF SNDLEN IS 0, THE FINANCE JOB *
       003690* WILL SIMPLY SEND AN INVITE TO THE WORK STATION TO ALLOW THE OPERATOR  *
       003700* TO INPUT THE NEXT TRANSACTION.                                        *
       003710*                                                                         *
       003720***************************************************************************
       003730
  256  003740     PERFORM CLEAN-UP.
       003750
       003760
       003770
       003780***************************************************************************
       003790*                                                                         *
       003800*THE PROCEDURE TO DO A VALID INQUIRY SENDS OUT THE DISPLAY IN TWO       *
       003810* RECORDS.                                                                *
       003820* THE OTS-MODE IS SET TO 1.  THIS PUTS THE 4704 DISPLAY INTO            *
       003830* LARGE SCREEN MODE WHEREBY 1920 CHARACTERS ARE DISPLAYED, THEREBY      *
       003840* ALLOWING MORE INFORMATION TO BE DISPLAYED.  THE FIRST RECORD HAS THE  *
       003850* BASIC CUSTOMER INFORMATION ACCOUNT, NAME, AND ADDRESS.  THE SECOND    *
       003860* HAS THE BALANCE INFORMATION.  BOTH SCREENS ARE SENT USING THE         *
       003870* "QFNWRT".                                                             *
       003880*                                                                         *
       003890***************************************************************************
       003900
6      003910 VALID-INQUIRY.
  257  003920     PERFORM BUILD-FORMAT-2.
  258  003930     CALL 'QFNWRT' USING WSID, SNDLEN, DATA-PARM, DATA-TYPE.
  259  003940     COMPUTE WRKBAL = TBAL   OF ACCOUNT-REC
       003950             - DLYWTH OF ACCOUNT-REC
       003960             + DLYDEP OF ACCOUNT-REC
       003970             + LIMIT1 OF ACCOUNT-REC
       003980             + LIMIT2 OF ACCOUNT-REC.
       003990
  260  004000     COMPUTE TOTRSV = RSVCS1 + RSVCS2 + RSVRM1 + RSVRM2.
  261  004010     PERFORM BUILD-FORMAT-3.
       004020
  262  004030     CALL 'QFNWRT' USING WSID, SNDLEN, DATA-PARM, DATA-TYPE.
       004040
       004050***************************************************************************
       004060*                                                                         *
       004070* SET SNDLEN BACK TO 0 SO THAT THE FINANCE JOB WILL NOT SEND ANY MORE   *
       004080* DATA FOR THIS TRANSACTION TO THE FINANCE FACILITY                     *
       004090*                                                                         *
       004100***************************************************************************
       004110
  263  004120     COMPUTE SNDLEN = 0.
       004130
       004140***************************************************************************
       004150*                                                                         *
       004160* BEFORE PERFORMING A WITHDRAWAL, MAKE SURE THE ACCOUNT IS STILL        *
```

Figure F-3 (Part 8 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      004170* ACTIVE.  IF IT IS NOT, PERFORM AN INQUIRY INSTEAD AND DISPLAY A       *
      004180* MESSAGE TO THE OPERATOR INDICATING THE ACCOUNT IS NO LONGER ACTIVE.   *
      004190*                                                                       *
      004200**************************************************************************
      004210
7     004220 ATTEMPT-WITHDRAWAL.
 264  004230     IF ACTIVE > 0
      004240     THEN
 265  004250       MOVE MSG(3) TO LCKDSC OF ACCOUNT-REC
 266  004260       PERFORM VALID-INQUIRY
      004270     ELSE
 267  004280       PERFORM START-WITHDRAWAL.
      004290
      004300**************************************************************************
      004310*                                                                       *
      004320* BEFORE PERFORMING A WITHDRAWAL, MAKE SURE THE ACCOUNT HAS SUFFICIENT  *
      004330* FUNDS.  IF IT DOES NOT, PERFORM AN INQUIRY INSTEAD AND DISPLAY A      *
      004340* MESSAGE TO THE OPERATOR INDICATING THERE ARE NOT SUFFICIENT FUNDS     *
      004350* FOR THE WITHDRAWAL.                                                   *
      004360*                                                                       *
      004370**************************************************************************
      004380
8     004390 START-WITHDRAWAL.
 268  004400     COMPUTE WRKBAL = TBAL   OF ACCOUNT-REC
      004410             - DLYWTH OF ACCOUNT-REC
      004420             + DLYDEP OF ACCOUNT-REC
      004430             + LIMIT1 OF ACCOUNT-REC
      004440             + LIMIT2 OF ACCOUNT-REC.
      004450
 269  004460     COMPUTE TOTRSV = RSVCS1 + RSVCS2 + RSVRM1 + RSVRM2.
 270  004470     COMPUTE TOTAVL = WRKBAL - TOTRSV.
      004480
 271  004490     IF AMOUNT > TOTAVL
      004500     THEN
 272  004510       MOVE MSG(4) TO LCKDSC OF ACCOUNT-REC
 273  004520       PERFORM VALID-INQUIRY
      004530     ELSE
 274  004540       PERFORM VALID-WITHDRAWAL.
      004550
      004560**************************************************************************
      004570*                                                                       *
      004580* THE PROCEDURE TO DO A VALID WITHDRAWAL SENDS OUT ONE RECORD.          *
      004590*                                                                       *
      004600**************************************************************************
      004610
9     004620 VALID-WITHDRAWAL.
 275  004630     ADD AMOUNT TO WTHDRL OF ACCOUNT-REC,
      004640            DLYWTH OF ACCOUNT-REC.
 276  004650     SUBTRACT AMOUNT FROM TBAL OF ACCOUNT-REC.
 277  004660     PERFORM BUILD-FORMAT-4.
 278  004670     CALL 'QFNWRT' USING WSID, SNDLEN, DATA-PARM, DATA-TYPE.
 279  004680     REWRITE ACCOUNT-REC.
      004690
      004700**************************************************************************
      004710* SET SNDLEN BACK TO 0 SO THAT THE FINANCE JOB WILL NOT SEND ANY MORE   *
```

Figure F-3 (Part 9 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      004720* DATA FOR THIS TRANSACTION TO THE FINANCE FACILITY                     *
      004730*************************************************************************
      004740
 280  004750     COMPUTE SNDLEN = 0.
      004760
      004770*************************************************************************
      004780*                                                                       *
      004790* AN ERROR MESSAGE WAS SENT TO THE FINANCE CONTROLLER.  THE MESSAGE     *
      004800* WILL BE WRITTEN AUTOMATICALLY BY THE FINANCE JOB UPON                 *
      004810* RETURN FROM THIS PROGRAM.                                             *
      004820*                                                                       *
      004830*************************************************************************
10    004840 BUILD-FORMAT-1.
 281  004850     MOVE ACCTNO TO ACCTNR OF FORMATTED-DATA-1.
 282  004860     MOVE 0 TO OTS-MODE.
 283  004870     MOVE NEWPAG TO NEW-PAGE-1.
 284  004880     MOVE DTASEP TO DATA-SEP-1.
 285  004890     MOVE FFS TO LASTFF-1
      004900
      004910*************************************************************************
      004920* SET MORE DATA FLAG TO 0 - THIS IS A COMPLETE RECORD -                 *
      004930* AND SET THE SEND LENGTH OF THE FORMATTED DATA + 16.                   *
      004940*************************************************************************
      004950
 286  004960     MOVE 0 TO MOREDT.
 287  004970     MOVE FORMATTED-DATA-1 TO OTS-FORMATTED-DATA.
 288  004980     MOVE OTS-OUTPUT-DATA TO DATA-PARM.
 289  004990     COMPUTE SNDLEN = 59.
      005000
11    005010 BUILD-FORMAT-2.
 290  005020     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-2.
            *       ** CORRESPONDING items for statement 290:
            *       **     ACCTNR
            *       **     NAME
            *       **     STR1
            *       **     STR2
            *       **     CITY
            *       **     OCUP
            *       ** End of CORRESPONDING items for statement 290
 291  005030     MOVE EDATE TO DATE-2.
 292  005040     MOVE 1 TO OTS-MODE.
 293  005050     MOVE NEWPAG TO NEW-PAGE-2.
 294  005060     MOVE NEWLIN TO NEWLIN-2B, NEWLIN-2C, NEWLIN-2D, NEWLIN-2E.
 295  005070     MOVE SETPOS TO SETPOS-2A, SETPOS-2C, SETPOS-2D, SETPOS-2E.
 296  005080     MOVE MOVHOR TO MOVHOR-2A, MOVHOR-2C, MOVHOR-2C, MOVHOR-2E.
 297  005090     MOVE HEX20 TO HEX20-2A.
 298  005100     MOVE HEX0F TO HEX0F-2C, HEX0F-2D, HEX0F-2E.
 299  005110     MOVE DTASEP TO DATA-SEP-2.
 300  005120     MOVE FFS TO LASTFF-2
      005130
      005140*************************************************************************
      005150*  SET THE SEND LENGTH TO LENGTH OF THE FORMATTED DATA + 16.            *
      005160*************************************************************************
      005170
 301  005180     MOVE 0 TO MOREDT.
```

Figure F-3 (Part 10 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
 302  005190     MOVE FORMATTED-DATA-2 TO OTS-FORMATTED-DATA.
 303  005200     MOVE OTS-OUTPUT-DATA TO DATA-PARM.
 304  005210     COMPUTE SNDLEN = 138.
      005220
12    005230 BUILD-FORMAT-3.
 305  005240     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-3.
            *       ** CORRESPONDING items for statement 305:
            *       **     TBAL
            *       **     DLYWTH
            *       **     DLYDEP
            *       **     LIMIT1
            *       **     LIMIT2
            *       **     LCKDSC
            *       ** End of CORRESPONDING items for statement 305
 306  005250     MOVE WRKBAL TO WRKBAL-3.
 307  005260     MOVE TOTRSV TO TOTRSV-3.
 308  005270     MOVE 1 TO OTS-MODE.
 309  005280     MOVE NEWLIN TO NEWLIN-3A, NEWLIN-3B, NEWLIN-3C, NEWLIN-3D,
      005290             NEWLIN-3E, NEWLIN-3F.
 310  005300     MOVE SETPOS TO SETPOS-3.
 311  005310     MOVE SETLIN TO SETLIN-3.
 312  005320     MOVE HEX06 TO HEX06-3
 313  005330     MOVE DTASEP TO DATA-SEP-3.
 314  005340     MOVE FFS TO LASTFF-3
      005350
      005360**************************************************************************
      005370* SET MORE DATA FLAG TO 0 - THIS TRANSACTION IS COMPLETE NOW -           *
      005380* AND SET THE SEND LENGTH TO LENGTH OF THE FORMATTED DATA + 16.          *
      005390**************************************************************************
      005400
 315  005410     MOVE 0 TO MOREDT.
 316  005420     MOVE FORMATTED-DATA-3 TO OTS-FORMATTED-DATA.
 317  005430     MOVE OTS-OUTPUT-DATA TO DATA-PARM.
 318  005440     COMPUTE SNDLEN = 208.
      005450
13    005460 BUILD-FORMAT-4.
 319  005470     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-4.
            *       ** CORRESPONDING items for statement 319:
            *       **     ACCTNR
            *       **     NAME
            *       **     OCUP
            *       ** End of CORRESPONDING items for statement 319
 320  005480     MOVE EDATE TO DATE-4.
 321  005490     MOVE AMOUNT TO AMOUNT-4.
 322  005500     MOVE WRKBAL TO WRKBAL-4.
 323  005510     MOVE TBAL OF ACCOUNT-REC TO BALOUT-4.
 324  005520     COMPUTE BIN-LINENBR = LINENBR + 1.
 325  005530     IF BIN-LINENBR > 18
 326  005540          COMPUTE BIN-LINENBR = 1.
 327  005550     MOVE 0 TO OTS-MODE.
 328  005560     MOVE NEWPAG TO NEWPAG-4.
 329  005570     MOVE NEWLIN TO NEWLIN-4B, NEWLIN-4C, NEWLIN-4D, NEWLIN-4E,
      005580             NEWLIN-4F, NEWLIN-4G, NEWLIN-4H, NEWLIN-4I.
 330  005590     MOVE SETPOS TO SETPOS-4A, SETPOS-4C.
 331  005600     MOVE MOVHOR TO MOVHOR-4A, MOVHOR-4C.
```

Figure F-3 (Part 11 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
 332  005610     MOVE DTASEP TO DATA-SEP-4A.
 333  005620     MOVE HEX0F TO HEX0F-4C.
 334  005630     MOVE HEX20 TO HEX20-4A.
 335  005640     MOVE FFS TO LASTFF-4
 336  005650     MOVE 0 TO MOREDT.
 337  005660     MOVE FORMATTED-DATA-4 TO OTS-FORMATTED-DATA.
 338  005670     MOVE OTS-OUTPUT-DATA TO DATA-PARM.
 339  005680     COMPUTE SNDLEN = 190.
      005690
      005700**************************************************************************
      005710*                                                                        *
      005720* TERMINATE PROGRAM.                                                     *
      005730*                                                                        *
      005740**************************************************************************
14    005750 CLEAN-UP.
      005760
 340  005770     CLOSE ACCOUNT.
      005780
 341  005790     STOP RUN.
      005800
                          * * * * *   E N D   O F   S O U R C E   * * * * *
STMT LVL  SOURCE NAME                   SECTION    DISP   LENGTH  TYPE   I-NAME    ATTRIBUTES
  17  FD  ACCOUNT                          FS                              .F01      DEVICE DATABASE, ORGANIZATION INDEXED,
                                                                                   ACCESS RANDOM  , BLOCK CONTAINS 188
                                                                                   CHARACTERS, RECORD CONTAINS 188
                                                                                   CHARACTERS, LABEL RECORDS STANDARD
  19  01  ACCOUNT-REC                      FS  00000000      188  GROUP  .D005DEE
  21  02  ACCOUNTR                         FS  00000000      188  GROUP  .D005F40
  22  03  ACCTNR                           FS  00000000        5  PACKED .D005FA8
  23  03  NAME                             FS  00000005       21  AN     .D006020
  24  03  STR1                             FS  00000026        3  AN     .D006084
  25  03  STR2                             FS  00000029       18  AN     .D0060E8
  26  03  CITY                             FS  00000047       16  AN     .D00614C
  27  03  ZIP                              FS  00000063        5  AN     .D0061B0
  28  03  OCUP                             FS  00000068       21  AN     .D006214
  29  03  TBAL                             FS  00000089        6  PACKED .D006278
  30  03  DLYWTH                           FS  00000095        6  PACKED .D0062F2
  31  03  DLYDEP                           FS  00000101        6  PACKED .D00636E
  32  03  LIMIT1                           FS  00000107        6  PACKED .D0063EA
  33  03  LIMIT2                           FS  00000113        6  PACKED .D006466
  34  03  RSVCS1                           FS  00000119        6  PACKED .D006548
  35  03  RSVCS2                           FS  00000125        6  PACKED .D0065C4
  36  03  RSVRM1                           FS  00000131        6  PACKED .D006640
  37  03  RSVRM2                           FS  00000137        6  PACKED .D0066BC
  38  03  ACTIVE                           FS  00000143        1  PACKED .D006738
  39  03  LCKDSC                           FS  00000144       38  AN     .D0067B0
  40  03  WTHDRL                           FS  00000182        6  PACKED .D006816
  42  01  FILLER                           WS  00000000        4  GROUP  .D006892
  43  02  FFFFFF                           WS  00000000        4  BINARY .D0068F8  VALUE
  44  02  FILLER                           WS  00000000        4  GROUP  .D006988  REDEFINES .D0068F8
  45  03  FILLER                           WS  00000000        1  AN     .D0069E6
  46  03  FFS                              WS  00000001        3  AN     .D006A44
  47  01  OTS-INPUT-DATA                   WS  00000000       98  GROUP  .D006AA8
  48  02  OTS-CTL-ITEMS                    WS  00000000       21  GROUP  .D006B16
  49  03  WSTYPE                           WS  00000000        2  AN     .D006B84
  50  03  CTLUNIT                          WS  00000002        2  AN     .D006BEA
  51  03  WSNO                             WS  00000004        2  AN     .D006C52
  52  03  AUDTNO                           WS  00000006        2  AN     .D006CB6
  53  03  TELLERNO                         WS  00000008        3  ZONED  .D006D1C
  54  03  LINENBR                          WS  00000011        2  ZONED  .D006D94
  55  03  TRNCDE                           WS  00000013        3  AN     .D006E0A
  56  03  SPLFNCT                          WS  00000016        1  AN     .D006E70
  57  03  RESVRD                           WS  00000017        3  AN     .D006ED8
  58  03  CTLUTYPE                         WS  00000020        1  AN     .D006F3E
```

Figure F-3 (Part 12 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
   59  02  OTS-DATA-AREA                 WS  00000021       77  GROUP  .D006FA6
   60  03  FLD1P                         WS  00000021        1  AN     .D007014
   61  03  FLD2P                         WS  00000022        1  AN     .D00707A
   62  03  FLD3P                         WS  00000023        1  AN     .D0070E0
   63  03  FLD4P                         WS  00000024        1  AN     .D007146
   64  03  FLD5P                         WS  00000025        1  AN     .D0071AC
   65  03  FLD6P                         WS  00000026        1  AN     .D007212
   66  03  FLD7P                         WS  00000027        1  AN     .D007278
   67  03  INPUT-FIELD                   WS  00000028       10  AN     .D0072DE  DIMENSION(7)
   68  IX  FLD                                                         .D00735C
   69  01  OTS-OUTPUT-DATA               WS  00000000      256  GROUP  .D0073C0
   70  02  OTS-CONTROL                   WS  00000000       16  GROUP  .D007430
   71  03  OTS-MODE                      WS  00000000        1  ZONED  .D00749C  VALUE
 STMT LVL  SOURCE NAME              SECTION      DISP   LENGTH  TYPE   I-NAME    ATTRIBUTES
   72  03  OTS-RSV                       WS  00000001        4  AN     .D007548  VALUE
   73  03  MOREDT                        WS  00000005        1  ZONED  .D0075C6  VALUE
   74  03  RSPDS                         WS  00000006        1  AN     .D00764E  VALUE
   75  03  DRSPF                         WS  00000007        1  AN     .D0076C8  VALUE
   76  03  RSPMS                         WS  00000008        1  AN     .D007742  VALUE
   77  03  MRSPF                         WS  00000009        1  AN     .D0077BC  VALUE
   78  03  RSPJP                         WS  00000010        1  AN     .D007836  VALUE
   79  03  JPRSPF                        WS  00000011        1  AN     .D0078B0  VALUE
   80  03  JPPARM                        WS  00000012        1  AN     .D00792A  VALUE
   81  03  RSPPS                         WS  00000013        1  AN     .D0079A4  VALUE
   82  03  PPRSPF                        WS  00000014        1  AN     .D007A1E  VALUE
   83  03  PSPARM                        WS  00000015        1  AN     .D007A98  VALUE
   84  02  OTS-FORMATTED-DATA            WS  00000016      240  AN     .D007B12
   85  01  FORMATTED-DATA-1              WS  00000000       43  GROUP  .D007B84
   86  02  NEW-PAGE-1                    WS  00000000        1  AN     .D007BF4
   87  02  TEXT-1                        WS  00000001       30  AN     .D007C5E
   88  02  ACCTNR                        WS  00000031        8  NE     .D007CC4
   89  02  DATA-SEP-1                    WS  00000039        1  AN     .D007D36
   90  02  LASTFF-1                      WS  00000040        3  AN     .D007DA0
   91  01  FORMATTED-DATA-2              WS  00000000      122  GROUP  .D007E08
   92  02  NEW-PAGE-2                    WS  00000000        1  AN     .D007E78
   93  02  SETPOS-2A                     WS  00000001        1  AN     .D007EE2
   94  02  MOVHOR-2A                     WS  00000002        1  AN     .D007F4C
   95  02  HEX20-2A                      WS  00000003        1  AN     .D007FB6
   96  02  DATE-2                        WS  00000004        8  NE     .D00801E
   97  02  NEWLIN-2B                     WS  00000012        1  AN     .D008098
   98  02  FILLER                        WS  00000013        5  AN     .D008102  VALUE
   99  02  ACCTNR                        WS  00000018        8  NE     .D008178
  100  02  FILLER                        WS  00000026        1  AN     .D0081EA
  101  02  NAME                          WS  00000027       21  AN     .D008248
  102  02  NEWLIN-2C                     WS  00000048        1  AN     .D0082A6
  103  02  SETPOS-2C                     WS  00000049        1  AN     .D008310
  104  02  MOVHOR-2C                     WS  00000050        1  AN     .D00837A
  105  02  HEX0F-2C                      WS  00000051        1  AN     .D0083E4
  106  02  STR1                          WS  00000052        3  AN     .D00844C
  107  02  STR2                          WS  00000055       18  AN     .D0084AA
  108  02  NEWLIN-2D                     WS  00000073        1  AN     .D008548
  109  02  SETPOS-2D                     WS  00000074        1  AN     .D0085B2
  110  02  MOVHOR-2D                     WS  00000075        1  AN     .D00861C
  111  02  HEX0F-2D                      WS  00000076        1  AN     .D008686
  112  02  CITY                          WS  00000077       16  AN     .D0086EE
  113  02  NEWLIN-2E                     WS  00000093        1  AN     .D00874C
  114  02  SETPOS-2E                     WS  00000094        1  AN     .D0087B6
  115  02  MOVHOR-2E                     WS  00000095        1  AN     .D008820
  116  02  HEX0F-2E                      WS  00000096        1  AN     .D00888A
  117  02  OCUP                          WS  00000097       21  AN     .D0088F2
  118  02  DATA-SEP-2                    WS  00000118        1  AN     .D008950
  119  02  LASTFF-2                      WS  00000119        3  AN     .D0089BA
  120  01  FORMATTED-DATA-3              WS  00000000      192  GROUP  .D008A22
  121  02  SETPOS-3                      WS  00000000        1  AN     .D008A92
  122  02  SETLIN-3                      WS  00000001        1  AN     .D008AFA
  123  02  HEX06-3                       WS  00000002        1  AN     .D008B62
  124  02  FILLER                        WS  00000003        5  AN     .D008BCA  VALUE
  125  02  TBAL                          WS  00000008       14  NE     .D008C40
  126  02  FILLER                        WS  00000022        5  AN     .D008CB8
```

Figure F-3 (Part 13 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
STMT LVL  SOURCE NAME                   SECTION    DISP   LENGTH  TYPE   I-NAME    ATTRIBUTES
 127  02  FILLER                           WS  00000027       10  AN     .D008D16  VALUE
 128  02  NEWLIN-3A                        WS  00000037        1  AN     .D008D90
 129  02  FILLER                           WS  00000038        5  AN     .D008DFA  VALUE
 130  02  DLYDEP                           WS  00000043       13  NE     .D008E70
 131  02  FILLER                           WS  00000056        3  AN     .D008EE8
 132  02  TOTRSV-3                         WS  00000059       14  NE     .D008F46
 133  02  NEWLIN-3B                        WS  00000073        1  AN     .D008FC8
 134  02  FILLER                           WS  00000074        5  AN     .D009032  VALUE
 135  02  DLYWTH                           WS  00000079       13  NE     .D0090A8
 136  02  NEWLIN-3C                        WS  00000092        1  AN     .D009120
 137  02  FILLER                           WS  00000093        5  AN     .D00918A  VALUE
 138  02  LIMIT1                           WS  00000098       13  NE     .D009200
 139  02  NEWLIN-3D                        WS  00000111        1  AN     .D009278
 140  02  FILLER                           WS  00000112        5  AN     .D0092E2  VALUE
 141  02  LIMIT2                           WS  00000117       13  NE     .D009358
 142  02  NEWLIN-3E                        WS  00000130        1  AN     .D0093D0
 143  02  FILLER                           WS  00000131        5  AN     .D00943A  VALUE
 144  02  WRKBAL-3                         WS  00000136       13  NE     .D0094B0
 145  02  NEWLIN-3F                        WS  00000149        1  AN     .D009548
 146  02  LCKDSC                           WS  00000150       38  AN     .D0095B2
 147  02  DATA-SEP-3                       WS  00000188        1  AN     .D009610
 148  02  LASTFF-3                         WS  00000189        3  AN     .D00967A
 149  01  FORMATTED-DATA-4                 WS  00000000      174  GROUP  .D0096E2
 150  02  NEWPAG-4                         WS  00000000        1  AN     .D009752
 151  02  SETPOS-4A                        WS  00000001        1  AN     .D0097BA
 152  02  MOVHOR-4A                        WS  00000002        1  AN     .D009824
 153  02  HEX20-4A                         WS  00000003        1  AN     .D00988E
 154  02  DATE-4                           WS  00000004        8  NE     .D0098F6
 155  02  NEWLIN-4B                        WS  00000012        1  AN     .D009970
 156  02  FILLER                           WS  00000013        5  AN     .D0099DA  VALUE
 157  02  ACCTNR                           WS  00000018        8  NE     .D009A50
 158  02  FILLER                           WS  00000026        1  AN     .D009AC2
 159  02  NAME                             WS  00000027       21  AN     .D009B20
 160  02  NEWLIN-4C                        WS  00000048        1  AN     .D009B7E
 161  02  SETPOS-4C                        WS  00000049        1  AN     .D009BE8
 162  02  MOVHOR-4C                        WS  00000050        1  AN     .D009C52
 163  02  HEX0F-4C                         WS  00000051        1  AN     .D009CBC
 164  02  OCUP                             WS  00000052       21  AN     .D009D24
 165  02  NEWLIN-4D                        WS  00000073        1  AN     .D009D82
 166  02  FILLER                           WS  00000074       10  AN     .D009DEC  VALUE
 167  02  AMOUNT-4                         WS  00000084       13  NE     .D009E64
 168  02  NEWLIN-4E                        WS  00000097        1  AN     .D009EE6
 169  02  NEWLIN-4F                        WS  00000098        1  AN     .D009F50
 170  02  FILLER                           WS  00000099       10  AN     .D009FBA  VALUE
 171  02  BALOUT-4                         WS  00000109       14  NE     .D00A034
 172  02  NEWLIN-4G                        WS  00000123        1  AN     .D00A0B6
 173  02  FILLER                           WS  00000124       10  AN     .D00A120  VALUE
 174  02  WRKBAL-4                         WS  00000134       14  NE     .D00A19A
 175  02  NEWLIN-4H                        WS  00000148        1  AN     .D00A21C
 176  02  NEWLIN-4I                        WS  00000149        1  AN     .D00A286
 177  02  TEXT-4                           WS  00000150       20  AN     .D00A2F0
 178  02  DATA-SEP-4A                      WS  00000170        1  AN     .D00A356
 179  02  LASTFF-4                         WS  00000171        3  AN     .D00A3C2
 180  01  DUMMY1                           WS  00000000       16  GROUP  .D00A42A
 181  02  BIN1                             WS  00000000        2  BINARY .D00A490  VALUE
```

Figure F-3 (Part 14 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
  STMT LVL  SOURCE NAME                   SECTION    DISP   LENGTH  TYPE   I-NAME    ATTRIBUTES
   182  02  BIN2                             WS  00000002       2  BINARY .D00A548  VALUE
   183  02  BIN3                             WS  00000004       2  BINARY .D00A5D2  VALUE
   184  02  BIN4                             WS  00000006       2  BINARY .D00A65C  VALUE
   185  02  BIN5                             WS  00000008       2  BINARY .D00A6E6  VALUE
   186  02  BIN6                             WS  00000010       2  BINARY .D00A770  VALUE
   187  02  BIN7                             WS  00000012       2  BINARY .D00A7FA  VALUE
   188  02  BIN-LINENBR                      WS  00000014       2  BINARY .D00A884
   189  01  DUMMY2                           WS  00000000      16  GROUP  .D00A900  REDEFINES .D00A42A
   190  02  NEWPAG                           WS  00000000       1  AN     .D00A966
   191  02  NEWLIN                           WS  00000001       1  AN     .D00A9CC
   192  02  MOVHOR                           WS  00000002       1  AN     .D00AA32
   193  02  SETPOS                           WS  00000003       1  AN     .D00AA98
   194  02  SETLIN                           WS  00000004       1  AN     .D00AAFE
   195  02  DTASEP                           WS  00000005       1  AN     .D00AB64
   196  02  HEX02                            WS  00000006       1  AN     .D00ABCA
   197  02  HEX12                            WS  00000007       1  AN     .D00AC30
   198  02  HEX0D                            WS  00000008       1  AN     .D00AC96
   199  02  HEX25                            WS  00000009       1  AN     .D00ACFC
   200  02  HEX0F                            WS  00000010       1  AN     .D00AD62
   201  02  HEX20                            WS  00000011       1  AN     .D00ADC8
   202  02  HEX0A                            WS  00000012       1  AN     .D00AE2E
   203  02  HEX06                            WS  00000013       1  AN     .D00AE94
   204  02  DUMMY00                          WS  00000014       1  AN     .D00AEFA
   205  02  LINENBR-HEX                      WS  00000015       1  AN     .D00AF62
   206  01  CONTROL-BYTE                     WS  00000000       1  GROUP  .D00AFCE
   207  02  DATA-TYPE                        WS  00000000       1  AN     .D00B03A  VALUE
   208  01  MESSAGES                         WS  00000000     120  GROUP  .D00B0B8
   209  02  MSG1                             WS  00000000      30  AN     .D00B120  VALUE
   210  02  MSG2                             WS  00000030      30  AN     .D00B1AC  VALUE
   211  02  MSG3                             WS  00000060      30  AN     .D00B238  VALUE
   212  02  MSG4                             WS  00000090      30  AN     .D00B2C6  VALUE
   213  01  MSG-TABLE                        WS  00000000     120  GROUP  .D00B358  REDEFINES .D00B0B8
   214  02  MSG                              WS  00000000      30  AN     .D00B3C2  DIMENSION(4)
   215  01  ACCOUNT-STATUS                   WS  00000000       1  AN     .D00B438
   216  88  VALID-ACCOUNT                    WS
   217  88  INVALID-ACCOUNT                  WS
   218  01  EDATE                            WS  00000000       6  ZONED  .D00B5CC
   219  01  ACCTNO                           WS  00000000       8  ZONED  .D00B644
   220  01  AMOUNT                           WS  00000000       7  PACKED .D00B6BC
   221  01  AMOUNT-IN                        WS  00000000      10  ZONED  .D00B736
   222  01  WRKBAL                           WS  00000000       6  PACKED .D00B7B0
   223  01  TOTRSV                           WS  00000000       6  PACKED .D00B82A
   224  01  TOTAVL                           WS  00000000       6  PACKED .D00B8A4
   226  01  WSID                             LS  00000001      10  AN     .D00B91E
   227  01  SNDLEN                           LS  00000002       8  PACKED .D00B982
   228  01  DATA-PARM                        LS  00000003     256  AN     .D00B9FE
    11      DB-FORMAT-NAME                   SR  00001076      10  AN     .D00BAD4
FILE SECTION uses 188 bytes of storage
WORKING-STORAGE SECTION uses 1086 bytes of storage
                    * * * * *   E N D   O F   D A T A   D I V I S I O N   M A P   * * * * *
  STMT
*  279  MSGID: LBL0412  SEVERITY: 20  SEQNBR:  004680
        Message . . . . :   INVALID KEY phrase not found in REWRITE
          statement. Accepted.
                        * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                          Message Summary
  Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
      1         0             0                1               0                  0
 Source records read . . . . . . . . :   579
 Copy records read . . . . . . . . . :   25
 Copy members processed  . . . . . . :   1
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   20
  LBL0901 00  Program OTSCBL1 created in library FNCLIB.
                       * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure F-3 (Part 15 of 15). COBOL/400 Program OTSCBL1 for Non-ICF Finance

RPG/400 Program Explanation

This section explains the RPG/400 account inquiry and withdrawal program example in Figure F-4 on page F-20.

1 This section of the program defines the database file (ACCOUNT).

ACCOUNT is the name of database file that contains the customer account information.

Note: The files used in this program are opened at the beginning of the ILE RPG/400 cycle.

2 OUTPUT is the name of the structure used with the special file function.

3 DATA is the name of the structure that contains the data parameters.

4 CNSTDS is the name of the structure that contains the control bytes used for screen control on the 4704.

5 OTSCTL is the name of the structure that contains the data to be sent to the controller.

6 The finance job passes three parameters to this program: the name of the device that sent the data, the data length, and the data.

7 This section of the program determines which fields in the data stream were sent by the controller.

8 This section of the program verifies that the account number exists. If the account number is not found, an error message is sent to the controller.

9 This section of the program determines the balance available to the customer. If an amount was sent from the controller that is less than the total amount available, a withdrawal is performed on the customer's account. Otherwise, the transaction is an account inquiry transaction.

10 This section of the program performs the end-of-program processing. The LR indicator is set to on and all files are closed implicitly. Then the program ends.

11 The output specifications are used together with the special file function. The data is formatted and passed to the program named in the special file function called (SUBEDT). This function then passes the data back during the read operation on file OUTPUT. In this example, the SUBEDT program passes back unchanged data.

```
Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . . . . . . . . :   FNCLIB/OTSRPG1
  Source file  . . . . . . . . . . :   FNCLIB/PGMSRC
  Source member  . . . . . . . . . :   OTSRPG1
  Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
  Source listing indentation . . . :   *NONE
  SAA flagging . . . . . . . . . . :   *NOFLAG
  Generation severity level  . . . :   9
  Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
  Replace program  . . . . . . . . :   *YES
  Target release . . . . . . . . . :   *CURRENT
  User profile . . . . . . . . . . :   *USER
  Authority  . . . . . . . . . . . :   *LIBCRTAUT
  Text . . . . . . . . . . . . . . :   *SRCMBRTXT
  Phase trace  . . . . . . . . . . :   *NO
  Intermediate text dump . . . . . :   *NONE
  Snap dump  . . . . . . . . . . . :   *NONE
  Codelist . . . . . . . . . . . . :   *NONE
  Ignore decimal data error  . . . :   *NO
Actual Program Source:
  Member . . . . . . . . . . . . . :   OTSRPG1
  File . . . . . . . . . . . . . . :   PGMSRC
  Library  . . . . . . . . . . . . :   FNCLIB
  Last Change  . . . . . . . . . . :   06/07/89  09:43:37
SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
                         S o u r c e   L i s t i n g
1         H                                                                                              *****
     100  FOUTPUT  CF  F     240            SPECIAL       SUBEDT
     200  FACCOUNT UF  E           K        DISK
     300  F*
     400  F********************************************************************                                  *
     500  F*                                                                                                     *
     600  F* THE SPECIAL FILE FUNCTION IN THIS PROGRAM IS USED TO EDIT                                           *
     700  F* THE OUTPUT FOR THE TELLER MACHINES.  IT ALLOWS THE USE OF                                           *
     800  F* RPG III OUTPUT SPECS FOR FORMATTING THE DATA.  WHEN AN OUTPUT                                       *
     900  F* TO THE SPECIAL FILE IS PERFORMED, THE DATA IS FORMATTED AND PASSED                                  *
    1000  F* TO THE USER-WRITTEN I/O ROUTING SPECIFIED IN THE SPECIAL FILE                                       *
    1100  F* STATEMENT (SUBEDT).  IN THIS EXAMPLE, SUBEDT ONLY DOES A                                            *
    1200  F* RETURN.  THE FORMATTED DATA IS THEN RETRIEVED WITH A READ FROM                                      *
    1300  F* THE SPECIAL FILE AND MOVED INTO THE DATA STRUCTURE SENT TO                                          *
    1400  F* THE 4700 CONTROL UNIT.                                                                              *
    1500  F*                                                                                                     *
    1600  E********************************************************************                                  *
    1700  E**
    1800  E** TABLES
    1900  E**
           RECORD FORMAT(S):  LIBRARY FNCLIB FILE ACCOUNT.
                    EXTERNAL FORMAT ACCOUNTR RPG NAME ACCOUNTR
    2000  E                    MSG     1   5 20               ERROR AND TEXT
    2100  E                    FLDS        7 10               OTS INPUT FLDS
2   2200  IOUTPUT  NF  01
    2300  I                                        1 240 OUTREC
 A000000   INPUT  FIELDS FOR RECORD ACCOUNTR FILE ACCOUNT FORMAT ACCOUNTR.
 A000001                                       P   1   50ACCTNR
 A000002                                           6  26 NAME
 A000003                                          27  29 STR1
 A000004                                          30  47 STR2
 A000005                                          48  63 CITY
 A000006                                          64  68 ZIP
 A000007                                          69  89 OCUP
 A000008                                       P  90  952TBAL
 A000009                                       P  96 1012DLYWTH
 A000010                                       P 102 1072DLYDEP
 A000011                                       P 108 1132LIMIT1
```

Figure F-4 (Part 1 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
 A000012                                       P 114 1192LIMIT2
 A000013                                       P 120 1252RSVCS1
 A000014                                       P 126 1312RSVCS2
 A000015                                       P 132 1372RSVRM1
 A000016                                       P 138 1432RSVRM2
 A000017                                       P 144 1440ACTIVE
 A000018                                         145 182 LCKDSC
 A000019                                       P 183 1882WTHDRL
3   2400  IDATA        DS
SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    2500  I                                        1 256 INPPRM
    2600  I                                       12  130LINENR
    2700  I                                       14  16 TRNCDE
    2800  I                                       22  22 FLD1P
    2900  I                                       23  23 FLD2P
    3000  I                                       24  24 FLD3P
    3100  I                                       25  25 FLD4P
    3200  I                                       26  26 FLD5P
    3300  I                                       27  27 FLD6P
    3400  I                                       28  28 FLD7P
    3500  I                                       29  98 FLDS
4   3600  ICNSTDS      DS
    3700  I                                        1   1 NEWPAG
    3800  I                                        2   2 NEWLIN
    3900  I                                        3   3 SETPOS
    4000  I                                        4   4 MOVHOR
    4100  I                                        5   5 SETLIN
    4200  I                                        6   6 DTASEP
    4300  I                                        7   7 X02
    4400  I                                        8   8 X08
    4500  I                                        9   9 X0A
    4600  I                                       10  10 X25
    4700  I                                       11  11 DTATYP
5   4800  IOTSCTL      DS                          1  16
    4900  I                                        1   10MODE
    5000  I                                        6   60MOREDT
    5100  I                                        7   7 RSPDS
    5200  I                                        8   8 DRSPF
    5300  I                                        9   9 RSPMS
    5400  I                                       10  10 MRSPF
    5500  I                                       11  11 RSPJP
    5600  I                                       12  12 JPRSPF
    5700  I                                       13  13 JPPARM
    5800  I                                       14  14 RSPPS
    5900  I                                       15  15 PPRSPF
    6000  I                                       16  16 PSPARM
    6100  C********************************************************************
    6200  C**                                                                 *
    6300  C** THE SBMFNCJOB INTERFACE PASSES THE PROGRAM THREE PARAMETERS:    *
    6400  C**      THE NAME OF THE REQUESTING TERMINAL                        *
    6500  C**      THE LENGTH OF THE DATA PLACED IN THE BUFFER                *
    6600  C**      THE DATA PLACED IN THE BUFFER                              *
    6700  C**                                                                 *
    6800  C********************************************************************
6   6900  C          *ENTRY    PLIST
    7000  C                    PARM           WSID   10        WORK STATION ID
    7100  C                    PARM           SNDLEN 155       OUTPUT LENGTH
    7200  C                    PARM           INPPR1256        INPUT DATA
    7300  C                    MOVELINPPR1    INPPRM           PUT IN WORK FLD
    7400  C                    BITOF'1'       NEWPAG           SET CONSTANTS
    7500  C                    BITOF'1'       NEWLIN           TO HEX'00'
    7600  C                    BITOF'1'       SETPOS           ''
```

Figure F-4 (Part 2 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
SEQUENCE                                                                          IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    7700  C                     BITOF'1'       MOVHOR
    7800  C                     BITOF'1'       SETLIN
    7900  C                     BITOF'1'       X02
    8000  C                     BITOF'1'       X08
    8100  C                     BITOF'1'       X0A
    8200  C                     BITOF'1'       X25
    8300  C                     BITON'45'      NEWPAG           SET SCREEN
    8400  C                     BITON'357'     NEWLIN           CONTROL
    8500  C                     BITON'235'     SETPOS           CONSTANTS
    8600  C                     BITON'4'       MOVHOR           ''
    8700  C                     BITON'5'       SETLIN
    8800  C                     BITON'0234567' DTASEP           ''
    8900  C                     BITON'4'       X08
    9000  C                     BITON'6'       X02              ''
    9100  C                     BITON'46'      X0A
    9200  C                     BITON'257'     X25
    9300  C**
    9400  C                     MOVEL'0'       DTATYP           SET 4700
    9500  C**                                                   CONTROL BYTE
    9600  C**                                                   TO INDICATE
    9700  C**                                              DATA FOLLOWS
    9800  C                     MOVE '0'       *IN99
    9900  C                     Z-ADD0         MODE
   10000  C                     Z-ADD0         MOREDT
   10100  C*********************************************************************
   10200  C**                                                                  *
   10300  C** OTS CAN PASS THE PROGRAM 7 INPUT FIELDS.  IT HAS 7 FLAGS         *
   10400  C** THAT INDICATE WHETHER THE INPUT FIELDS WERE ENTERED              *
   10500  C**     FLD1P CONTAINS "1" IF FIELD 1 WAS ENTERED, " " OTHERWISE     *
   10600  C**     FLD2P CONTAINS "2" IF FIELD 2 WAS ENTERED, " " OTHERWISE     *
   10700  C**                                                                  *
   10800  C** THE DATA IS LOADED SEQUENTIALLY INTO THE BUFFER.                 *
   10900  C** IF THE OPERATOR ENTERS FIELDS 1, 3, 5, 7 THEY WILL BE            *
   11000  C** IN THE INPUT FIELDS IN POSITIONS 1, 2, 3, AND 4                  *
   11100  C**                                                                  *
   11200  C** INSPECT THE INPUT FIELDS AND MOVE THE INPUT DATA FIELDS          *
   11300  C** INTO THE CORRECT PROGRAM FIELDS                                  *
   11400  C**                                                                  *
   11500  C*********************************************************************
   11600  C**
   11700  C** MOVE THE INPUT FIELDS TO THE CORRECT PROGRAM FIELDS
   11800  C**
   11900  C                     Z-ADD1         I       20
   12000  C*
7  12100  C*  GET FIELD 1 - THE ACCOUNT NUMBER
   12200  C*
   12300  C           FLD1P     IFEQ '1'                                                B001
   12400  C                     MOVE FLDS,I    ACCTNR  80                                001
   12500  C                     ADD  1         I                                         001
   12600  C                     ELSE                                                    X001
   12700  C                     Z-ADD0         ACCTNR                                    001
   12800  C                     END                                                     E001
   12900  C*
   13000  C*  GET FIELD 2 - THE TRANSACTION AMOUNT
```

Figure F-4 (Part 3 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   13100  C*
   13200  C           FLD2P     IFEQ '2'                                               B001
   13300  C                     MOVE FLDS,I    AMNT1  102                               001
   13400  C                     ADD  1         I                                        001
   13500  C                     ELSE                                                   X001
   13600  C                     Z-ADD0         AMNT1                                    001
   13700  C                     END                                                    E001
   13800  C*
   13900  C*  IGNORE FIELD 3 IF IT IS PRESENT ... NOT USED IN THIS PROGRAM
   14000  C*
   14100  C           FLD3P     IFEQ '3'                                               B001
   14200  C                     ADD  1         I                                        001
   14300  C                     END                                                    E001
   14400  C********************************************************************
   14500  C**                                                                 *
   14600  C** FIRST MAKE SURE THAT AN ACCOUNT NUMBER WAS ENTERED.             *
   14700  C**     IF NO ACCOUNT NUMBER OR AN ACCOUNT NUMBER OF ZERO           *
   14800  C**     WAS ENTERED, SEND AN ERROR MESSAGE BACK TO THE OPERATOR.    *
   14900  C**                                                                 *
   15000  C**     IF AN ACCOUNT NUMBER IS PRESENT, CHAIN OUT TO THE ACCOUNT   *
   15100  C**     FILE TO GET THE MASTER RECORD.                              *
   15200  C**                                                                 *
   15300  C**     IF THE CHAIN FAILS, SEND AN ERROR MESSAGE BACK TO THE       *
   15400  C**     OPERATOR.                                                   *
   15500  C**                                                                 *
   15600  C********************************************************************
8  15700  C           ACCTNR    IFEQ *ZERO                                             B001
   15800  C                     MOVELMSG,4     TEXT   20                                001
   15900  C                     EXSR OUTP01                                             001
   16000  C                     ELSE                                                   X001
   16100  C           ACCTNR    CHAINACCOUNT              90                    1       001
   16200  C           *IN90     IFEQ '1'                                               B002
   16300  C                     MOVE MSG,1     TEXT                                     002
   16400  C                     EXSR OUTP01                                             002
   16500  C                     ELSE                                                   X002
   16600  C********************************************************************
   16700  C**                                                                 *
   16800  C** CHECK TO MAKE SURE THAT THE ACCOUNT IS ACTIVE.                  *
   16900  C**    IF IT IS NOT, MAKE AMNT1 ZERO...MEANING THAT TRANSACTION     *
   17000  C**    IS AN INQUIRY.  ALSO SEND A MESSAGE TO THE SCREEN            *
   17100  C**    INDICATING THAT THE ACCOUNT IS INACTIVE.                     *
   17200  C**                                                                 *
   17300  C********************************************************************
9  17400  C           ACTIVE    IFGT 0                                                 B003
   17500  C                     Z-ADD0         AMNT1                                    003
   17600  C                     MOVE MSG,3     TEXT                                     003
   17700  C                     END                                                    E003
   17800  C                     MOVE TBAL      BALOUT 102                               002
   17900  C                     ADD  DLYWTH    BALOUT                                   002
   18000  C                     ADD  DLYDEP    BALOUT                                   002
   18100  C           BALOUT    ADD  LIMIT1    WRKBAL 102                               002
   18200  C                     ADD  LIMIT2    WRKBAL                                   002
   18300  C                     MOVELMSG,5     FELD4  12                                002
   18400  C                     ADD  RSVCS1    TOTRSV 102                               002
```

Figure F-4 (Part 4 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   18500  C                     ADD  RSVCS2    TOTRSV                                   002
   18600  C                     ADD  RSVRM1    TOTRSV                                   002
   18700  C                     ADD  RSVRM2    TOTRSV                                   002
   18800  C                     MOVE DLYDEP    DEPSIO 102                               002
   18900  C                     MOVE DLYWTH    WTHDRO 102                               002
   19000  C           AMNT1     IFEQ 0                                                 B003
   19100  C                     EXSR OUTP02                                             003
   19200  C                     ELSE                                                   X003
   19300  C           TBAL      IFLT 0                                                 B004
   19400  C                     Z-SUBBALOUT    BALMGS 102                               004
   19500  C                     ELSE                                                   X004
   19600  C                     Z-ADDBALOUT    BALMGS                                   004
   19700  C                     END                                                    E004
   19800  C                     MOVE WRKBAL    TOTAVL 102                               003
   19900  C                     SUB  TOTRSV    TOTAVL                                   003
   20000  C           AMNT1     IFGT TOTAVL                                            B004
   20100  C                     MOVE MSG,2     TEXT                                     004
   20200  C                     EXSR OUTP01                                             004
   20300  C                     ELSE                                                   X004
   20400  C                     ADD  AMNT1     WTHDRL                                   004
   20500  C                     SUB  AMNT1     DLYWTH                                   004
   20600  C                     SUB  AMNT1     BALOUT                                   004
   20700  C                     SUB  AMNT1     BALMGS                                   004
   20800  C                     MOVELNAME      NAME1  20                                004
   20900  C                     UPDATACCOUNTR                                           004
   21000  C                     EXSR OUTP03                                             004
   21100  C                     END                                                    E004
   21200  C                     END                                                    E003
   21300  C                     END                                                    E002
   21400  C                     END                                                    E001
   21500  C** END OF JOB
   21600  C**
10 21700  C                     SETON                         LR                    3
   21800  C                     RETRN
   21900  C********************************************************************
   22000  C** IF YOU WILL SEND JUST ONE RECORD, YOU DO NOT NEED               *
   22100  C** THE 'QFNWRT' PROGRAM.                                           *
   22200  C********************************************************************
   22300  C           OUTP01    BEGSR
   22400  C                     Z-ADD0         MOREDT
   22500  C                     EXCPTOUTPD1
   22600  C                     READ OUTPUT                   99                    3
   22700  C                     MOVE OUTREC    INPPR1
   22800  C                     MOVELOTSCTL    INPPR1
   22900  C                     Z-ADD46        SNDLEN
   23000  C                     ENDSR
   23100  C*
   23200  C           OUTP02    BEGSR
   23300  C                     Z-ADD0         MOREDT
   23400  C                     Z-ADD1         MODE
   23500  C                     EXCPTOUTPD2
   23600  C                     READ OUTPUT                   99                    3
   23700  C                     MOVE OUTREC    INPPR1
   23800  C                     MOVELOTSCTL    INPPR1
```

Figure F-4 (Part 5 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   23900  C                     Z-ADD208       SNDLEN
   24000  C                     CALL 'QFNWRT'
   24100  C                     PARM           WSID
   24200  C                     PARM           SNDLEN
   24300  C                     PARM           INPPR1
   24400  C                     PARM           DTATYP
   24500  C                     Z-ADD0         MOREDT
   24600  C                     EXCPTOUTPD3
   24700  C                     READ OUTPUT                   99                    3
   24800  C                     MOVE OUTREC    INPPR1
   24900  C                     MOVELOTSCTL    INPPR1
   25000  C                     Z-ADD85        SNDLEN
   25100  C                     ENDSR
   25200  C*
   25300  C           OUTP03    BEGSR
   25400  C                     Z-ADD0         MOREDT
   25500  C           LINENR    ADD  1         LINENR
   25600  C           LINENR    IFGT 16                                                B001
   25700  C                     Z-ADD1         LINENR                                   001
   25800  C                     END                                                    E001
   25900  C                     EXCPTOUTPD4
   26000  C                     READ OUTPUT                   99                    3
   26100  C                     MOVE OUTREC    INPPR1
   26200  C                     MOVELOTSCTL    INPPR1
   26300  C                     Z-ADD191       SNDLEN
   26400  C                     ENDSR
   26500  C********************************************************************
   26600  C**                                                                 *
   26700  C** OUTPUT TO FORMAT ERROR DISPLAY FOR INVALID OR                  *
   26800  C** NO ACCOUNT NUMBER ENTERED                                       *
   26900  C**                                                                 *
   27000  C********************************************************************
11 27100  OOUTPUT  E                OUTPD1
   27200  O                         NEWPAG
   27300  O                         TEXT
   27400  O                         DTASEP    30
   27500  O********************************************************************
   27600  O**                                                                 *
   27700  O** OUTPUT TO FORMAT FIRST PART OF INQUIRY DISPLAY                  *
   27800  O**                                                                 *
   27900  O********************************************************************
   28000  O        E                OUTPD2
   28100  O                         NEWPAG
   28200  O                         TEXT
   28300  O                         NEWLIN
   28400  O                                      'ACCONTNR'
   28500  O                         ACCTNRZ +  1
   28600  O                         NEWLIN
   28700  O                                      'BALANCE'
   28800  O                         BALOUTJ +  1
   28900  O                         NEWLIN
   29000  O                                      'DEP'
```

Figure F-4 (Part 6 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   29100  O                         DEPSIO2 +  1
   29200  O                                 +  2 'WTH'
   29300  O                         WTHDRO2 +  1
   29400  O                         NEWLIN
   29500  O                                      'RES AMOUNT'
   29600  O                         TOTRSV4 +  3
   29700  O                         NEWLIN
   29800  O                                      'LIM1'
   29900  O                         LIMIT12 +  4
   30000  O                                 +  2 'LIM2'
   30100  O                         LIMIT22 +  4
   30200  O                         NEWLIN
   30300  O                         FELD4
   30400  O                         WRKBALJ +  1
   30500  O                         NEWLIN
   30600  O                         LCKDSC
   30700  O                         DTASEP   192
   30800  O*********************************************************************
   30900  O**                                                                  *
   31000  O** OUTPUT TO FORMAT SECOND PART OF INQUIRY DISPLAY                  *
   31100  O**                                                                  *
   31200  O*********************************************************************
   31300  O        E                OUTPD3
   31400  O                         SETPOS
   31500  O                         SETLIN
   31600  O                         X08
   31700  O                         NAME
   31800  O                         NEWLIN
   31900  O                         STR1
   32000  O                         STR2
   32100  O                         NEWLIN
   32200  O                         ZIP
   32300  O                         CITY    +  1
   32400  O                         DTASEP    69
   32500  O*********************************************************************
   32600  O**                                                                  *
   32700  O** OUTPUT TO FORMAT WITHDRAWAL DISPLAY                              *
   32800  O**                                                                  *
   32900  O*********************************************************************
   33000  O        E                OUTPD4
   33100  O                         NEWPAG     1
   33200  O                         NAME
   33300  O                         SETPOS
   33400  O                         MOVHOR
   33500  O                         X0A
   33600  O                         UDATE Y
   33700  O                         NEWLIN
   33800  O                         OCUP
   33900  O                         NEWLIN
   34000  O                                      'ACCOUNT-NR'
   34100  O                         ACCTNRZ +  1
   34200  O                         NEWLIN
   34300  O                                      'AMOUNT'
   34400  O                         AMNT1 2
```

Figure F-4 (Part 7 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
 SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
 NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    34500  O                         NEWLIN
    34600  O                         NEWLIN
    34700  O                                      'BALANCE'
    34800  O                         BALOUTJ +  6
    34900  O                         NEWLIN
    35000  O                         FELD4
    35100  O                         WRKBALJ +  1
    35200  O                         NEWLIN
    35300  O                         NEWLIN
    35400  O                         TEXT
    35500  O                         DTASEP   175
    35600  O*********************************************************************
  B000000   OUTPUT FIELDS FOR RECORD ACCOUNTR FILE ACCOUNT FORMAT ACCOUNTR.
  B000001                            ACCTNR     5P PACK  8,0
  B000002                            NAME      26  CHAR   21
  B000003                            STR1      29  CHAR    3
  B000004                            STR2      47  CHAR   18
  B000005                            CITY      63  CHAR   16
  B000006                            ZIP       68  CHAR    5
  B000007                            OCUP      89  CHAR   21
  B000008                            TBAL      95P PACK 10,2
  B000009                            DLYWTH   101P PACK 10,2
  B000010                            DLYDEP   107P PACK 10,2
  B000011                            LIMIT1   113P PACK 10,2
  B000012                            LIMIT2   119P PACK 10,2
  B000013                            RSVCS1   125P PACK 10,2
  B000014                            RSVCS2   131P PACK 10,2
  B000015                            RSVRM1   137P PACK 10,2
  B000016                            RSVRM2   143P PACK 10,2
  B000017                            ACTIVE   144P PACK  1,0
  B000018                            LCKDSC   182  CHAR   38
  B000019                            WTHDRL   188P PACK 10,2
           * * * * *   E N D   O F   S O U R C E   * * * * *
          A d d i t i o n a l   D i a g n o s t i c   M e s s a g e s
SEQUENCE                                                                                             LAST
 NUMBER    *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8          UPDATE
                     C o m p i l e - T i m e   T a b l e s
 Table/Array  . . . . . . :  MSG
    35800  ACCOUNT-NR NOT FOUND
    35900  INSUFFICIENT FUNDS
    36000  ACCOUNT INACTIVE
    36100  NO ACCOUNT-NR ENTERED
    36200  WORK LIMIT
 TABLE OF END POSITION OFFSETS FOR FIELDS DESCRIBED USING POSITION NOTATION.
  STMT NO     POS     STMT NO     POS     STMT NO     POS     STMT NO     POS
    27200       1       27300      21       28100       1       28200      21
    28300      22       28400      30       28500      39       28600      40
    28700      47       28800      62       28900      63       29000      66
    29100      80       29200      85       29300      99       29400     100
    29500     110       29600     124       29700     125       29800     129
    29900     146       30000     152       30100     169       30200     170
    30300     182       30400     197       30500     198       30600     236
    31400       1       31500       2       31600       3       31700      24
    31800      25       31900      28       32000      46       32100      47
    32200      52       32300      69       33200      22       33300      23
    33400      24       33500      25       33600      33       33700      34
    33800      55       33900      56       34000      66       34100      75
    34200      76       34300      82       34400      95       34500      96
    34600      97       34700     104       34800     124       34900     125
    35000     137       35100     152       35200     153       35300     154
    35400     174
```

Figure F-4 (Part 8 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
                   K e y   F i e l d   I n f o r m a t i o n
                       PHYSICAL    LOGICAL
        FILE/RCD       FIELD       FIELD       ATTRIBUTES
    02  ACCOUNT
             ACCOUNTR
                       ACCTNR                  PACK  8,0  SIGNED
                         C r o s s   R e f e r e n c e
 File and Record References:
        FILE/RCD    DEV/RCD     REFERENCES (D=DEFINED)
    02  ACCOUNT     DISK            200D   16100
          ACCOUNTR                  200D A000000    20900  B000000
    01  OUTPUT      SPECIAL         100D    2200    22600    23600    24700
                                  26000    27100    28000    31300    33000
 Field References:
        FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
        *ENTRY      PLIST      6900D
        *IN90       A(1)      16200
        *IN99       A(1)       9800M
        ACCTNR      P(8,0)  A000001D   12400D   12700M   15700    16100
                              28500    34100  B000001D
        ACTIVE      P(1,0)  A000017D   17400  B000017D
        AMNT1       P(10,2)   13300D   13600M   17500M   19000    20000
                              20400    20500    20600    20700    34400
        BALMGS      P(10,2)   19400D   19600M   20700M
        BALOUT      P(10,2)   17800D   17900M   18000M   18100    19400
                              19600    20600M   28800    34800
        CITY        A(16)   A000005D   32300  B000005D
* 7031  CNSTDS      DS(11)     3600D
* 7031  DATA        DS(256)    2400D
        DEPSIO      P(10,2)   18800D   29100
        DLYDEP      P(10,2) A000010D   18000    18800  B000010D
        DLYWTH      P(10,2) A000009D   17900    18900    20500M B000009D
* 7031  DRSPF       A(1)       5200D
        DTASEP      A(1)       4200D    8800M   27400    30700    32400
                              35500
        DTATYP      A(1)       4700D    9400M   24400
        FELD4       A(12)     18300D   30300    35000
        FLDS(7)     A(10)      2100D    3500D
          FLDS,I              12400    13300
        FLD1P       A(1)       2800D   12300
        FLD2P       A(1)       2900D   13200
        FLD3P       A(1)       3000D   14100
* 7031  FLD4P       A(1)       3100D
* 7031  FLD5P       A(1)       3200D
* 7031  FLD6P       A(1)       3300D
* 7031  FLD7P       A(1)       3400D
        I           P(2,0)    11900D   12400    12500M   13300    13400M
                              14200M
        INPPRM      A(256)     2500D    7300M
        INPPR1      A(256)     7200D    7300    22700M   22800M   23700M
                              23800M   24300    24800M   24900M   26100M
```

Figure F-4 (Part 9 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
                              26200M
* 7031  JPPARM      A(1)       5700D
* 7031  JPRSPF      A(1)       5600D
        LCKDSC      A(38)   A000018D   30600  B000018D
        LIMIT1      P(10,2) A000011D   18100    29900  B000011D
        LIMIT2      P(10,2) A000012D   18200    30100  B000012D
        LINENR      Z(2,0)     2600D   25500    25500M   25600    25700M
        MODE        Z(1,0)     4900D    9900M   23400M
        MOREDT      Z(1,0)     5000D   10000M   22400M   23300M   24500M
                              25400M
        MOVHOR      A(1)       4000D    7700M    8600M   33400
* 7031  MRSPF       A(1)       5400D
        MSG(5)      A(20)      2000D
          MSG,1               16300
          MSG,2               20100
          MSG,3               17600
          MSG,4               15800
          MSG,5               18300
        NAME        A(21)   A000002D   20800    31700    33200  B000002D
        NAME1       A(20)     20800D
        NEWLIN      A(1)       3800D    7500M    8400M   28300    28600
                              28900    29400    29700    30200    30500
                              31800    32100    33700    33900    34200
                              34500    34600    34900    35200    35300
        NEWPAG      A(1)       3700D    7400M    8300M   27200    28100
                              33100
        OCUP        A(21)   A000007D   33800  B000007D
        OTSCTL(1)   DS(16)     4800D   22800    23800    24900    26200
        OUTPD1      EXCPT      22500    27100
        OUTPD2      EXCPT      23500    28000
        OUTPD3      EXCPT      24600    31300
        OUTPD4      EXCPT      25900    33000
        OUTP01      BEGSR      15900    16400    20200    22300D
        OUTP02      BEGSR      19100    23200D
        OUTP03      BEGSR      21000    25300D
        OUTREC      A(240)     2300D   22700    23700    24800    26100
* 7031  PPRSPF      A(1)       5900D
* 7031  PSPARM      A(1)       6000D
* 7031  RSPDS       A(1)       5100D
* 7031  RSPJP       A(1)       5500D
* 7031  RSPMS       A(1)       5300D
* 7031  RSPPS       A(1)       5800D
        RSVCS1      P(10,2) A000013D   18400  B000013D
        RSVCS2      P(10,2) A000014D   18500  B000014D
        RSVRM1      P(10,2) A000015D   18600  B000015D
        RSVRM2      P(10,2) A000016D   18700  B000016D
        SETLIN      A(1)       4100D    7800M    8700M   31500
        SETPOS      A(1)       3900D    7600M    8500M   31400    33300
        SNDLEN      P(15,5)    7100D   22900M   23900M   24200    25000M
                              26300M
        STR1        A(3)    A000003D   31900  B000003D
        STR2        A(18)   A000004D   32000  B000004D
        TBAL        P(10,2) A000008D   17800    19300  B000008D
        TEXT        A(20)     15800D   16300M   17600M   20100M   27300
                              28200    35400
        TOTAVL      P(10,2)   19800D   19900M   20000
        TOTRSV      P(10,2)   18400D   18500M   18600M   18700M   19900
```

Figure F-4 (Part 10 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
                              29600
* 7031  TRNCDE      A(3)       2700D
        UDATE       P(6,0)    33600
        WRKBAL      P(10,2)   18100D   18200M   19800    30400    35100
        WSID        A(10)      7000D   24100
        WTHDRL      P(10,2) A000019D   20400M B000019D
        WTHDRO      P(10,2)   18900D   29300
        X0A         A(1)       4500D    8100M    9100M   33500
        X02         A(1)       4300D    7900M    9000M
        X08         A(1)       4400D    8000M    8900M   31600
        X25         A(1)       4600D    8200M    9200M
        ZIP         A(5)    A000006D   32200  B000006D
        *ZERO       LITERAL   15700
        'QFNWRT'    LITERAL   24000
        '0'         LITERAL    9400     9800
        '0234567'   LITERAL    8800
        '1'         LITERAL    7400     7500     7600     7700     7800
                               7900     8000     8100     8200    12300
                              16200
        '2'         LITERAL   13200
        '235'       LITERAL    8500
        '257'       LITERAL    9200
        '3'         LITERAL   14100
        '357'       LITERAL    8400
        '4'         LITERAL    8600     8900
        '45'        LITERAL    8300
        '46'        LITERAL    9100
        '5'         LITERAL    8700
        '6'         LITERAL    9000
        0           LITERAL    9900    10000    12700    13600    17400
                              17500    19000    19300    22400    23300
                              24500    25400
        1           LITERAL   11900    12500    13400    14200    16300
                              23400    25500    25700
        16          LITERAL   25600
        191         LITERAL   26300
        2           LITERAL   20100
        208         LITERAL   23900
        3           LITERAL   17600
        4           LITERAL   15800
        46          LITERAL   22900
        5           LITERAL   18300
        85          LITERAL   25000
 Indicator References:
        INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
        *IN           9800M   16200
        LR           21700M
* 7031  01            2200M
        90           16100M   16200
        99            9800M   22600M   23600M   24700M   26000M
      * * * * *   E N D   O F   C R O S S   R E F E R E N C E   * * * * *
                        M e s s a g e   S u m m a r y
* QRG7031 Severity:  00   Number:   18
          Message . . . . :   The Name or indicator is not referenced.
      * * * * *   E N D   O F   M E S S A G E   S U M M A R Y   * * * * *
                          F i n a l   S u m m a r y
 Message Count:  (by Severity Number)
            TOTAL    00     10     20     30     40     50
              18      18      0      0      0      0      0
 Program Source Totals:
    Records . . . . . . . . . . :   362
    Specifications  . . . . . . :   249
    Table Records . . . . . . . :   5
    Comments  . . . . . . . . . :   107
 PRM has been called.
 Program OTSRPG1 is placed in library FNCLIB. 00 highest Error-Severity-Code.
          * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure F-4 (Part 11 of 11). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . . . . . . . . :   FNCLIB/SUBEDT
  Source file  . . . . . . . . . . :   FNCLIB/PGMSRC
  Source member  . . . . . . . . . :   SUBEDT
  Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
  Source listing indentation . . . :   *NONE
  SAA flagging . . . . . . . . . . :   *NOFLAG
  Generation severity level  . . . :   9
  Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
  Replace program  . . . . . . . . :   *YES
  Target release . . . . . . . . . :   *CURRENT
  User profile . . . . . . . . . . :   *USER
  Authority  . . . . . . . . . . . :   *LIBCRTAUT
  Text . . . . . . . . . . . . . . :   *SRCMBRTXT
  Phase trace  . . . . . . . . . . :   *NO
  Intermediate text dump . . . . . :   *NONE
  Snap dump  . . . . . . . . . . . :   *NONE
  Codelist . . . . . . . . . . . . :   *NONE
  Ignore decimal data error  . . . :   *NO
Actual Program Source:
  Member . . . . . . . . . . . . . :   SUBEDT
  File . . . . . . . . . . . . . . :   PGMSRC
  Library  . . . . . . . . . . . . :   FNCLIB
  Last Change  . . . . . . . . . . :   06/07/89  09:39:30
SEQUENCE                                                                         IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
                         S o u r c e   L i s t i n g
     100  F*****************************************************************                  04/22/88
     200  F*                                                                                  04/22/88
     300  F*  THIS SPECIAL FILE ACCEPTS DATA FROM THE OTSRPG1 PROGRAM                         06/07/89
     400  F*  AND FORMATS THE DATA FOR OUTPUT TO THE THE 4700 FINANCE                         06/07/89
     500  F*  DEVICES.  WHEN A READ FROM THIS SPECIAL FILE IS PERFORMED,                      06/07/89
     600  F*  THE FORMATTED DATA IS RETURNED TO THE RPG PROGRAM.                              06/07/89
     700  F*                                                                                  04/22/88
     800  F*****************************************************************                  04/22/88
          H                                                                                            *****
     900  C           *ENTRY    PLIST                                                         04/22/88
    1000  C                     PARM           OPTCDE  1                                      04/22/88
    1100  C                     PARM           RTNSTS  1                                      04/22/88
    1200  C                     PARM           ERRFND  50                                     04/22/88
    1300  C                     PARM           RECORD240                                      04/22/88
    1400  C                     RETRN                                                         04/22/88
    1500  ******************************************************************                  04/22/88
    1600  ******************************************************************                  04/22/88
          * * * * *   E N D   O F   S O U R C E   * * * * *
         A d d i t i o n a l   D i a g n o s t i c   M e s s a g e s
                        C r o s s   R e f e r e n c e
Field References:
       FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
       *ENTRY      PLIST    900D
       ERRFND      P(5,0)  1200D
       OPTCDE      A(1)    1000D
       RECORD      A(240)  1300D
       RTNSTS      A(1)    1100D
     * * * * *   E N D   O F   C R O S S   R E F E R E N C E   * * * * *
                          F i n a l   S u m m a r y
No errors found in source program.
Program Source Totals:
   Records . . . . . . . . . . :   16
   Specifications  . . . . . . :   6
   Table Records . . . . . . . :   0
   Comments  . . . . . . . . . :   10
PRM has been called.
Program SUBEDT is placed in library FNCLIB. 00 highest Error-Severity-Code.
         * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure F-5. ILE RPG/400 Program SUBEDT for Non-ICF Finance

Appendix G. Configuration Examples

This appendix provides configuration examples for Financial Branch System Services (FBSS) finance controllers attached to AS/400 systems using SDLC, token-ring, and X.25 network lines. Each example shows the CL commands used to create the AS/400 configuration objects and the FBSS configuration displays containing values that must match the AS/400 configuration.

Configuration Example for FBSS Controller Using SDLC

The following CL commands can be used to create the AS/400 configuration objects needed to attach an FBSS controller with three attached devices to an AS/400 system over a nonswitched SDLC line.

```
CRTLINSDLC LIND(FBSSLINE) RSRCNAME(LIN071) ROLE(*PRI)
           CNN(*NONSWTPP) NRZI(*YES) DUPLEX(*HALF)
CRTCTLFNC  CTLD(CTLFBSS) TYPE(*FBSS) MODEL(0)
           LINKTYPE(*SDLC) SWITCHED(*NO)
           LINE(FBSSLINE) EXCHID(05712345)
           STNADR(C1)
CRTDEVFNC  DEVD(FBSSDEV02) TYPE(*FNCICF) LOCADR(02)
           RMTLOCNAME(DEV02) CTL(CTLFBSS)
CRTDEVFNC  DEVD(FBSSDEV03) TYPE(*FNCICF) LOCADR(03)
           RMTLOCNAME(DEV03) CTL(CTLFBSS)
CRTDEVFNC  DEVD(FBSSDEV04) TYPE(*FNCICF) LOCADR(04)
           RMTLOCNAME(DEV04) CTL(CTLFBSS)
```

Program Explanation

The following displays are used to configure the FBSS controller for SDLC communications with the AS/400 system. Values that must match the AS/400 configuration are described below.

1 The *Data Link Control* specified on the FBSS Communication Servers display (*SDLC*) must match the link type (LINKTYPE parameter) specified on the CRTCTLFNC command (see Figure G-1).

2 If an SSCP name is specified for the FBSS controller, the value must match the AS/400 SSCP identifier (SSCPID parameter) specified on the CRTCTLFNC command (see Figure G-2 on page G-2).

3 FBSS *Station address* must match the station address (STNADR parameter) specified on the CRTCTLFNC command (see Figure G-3 on page G-2).

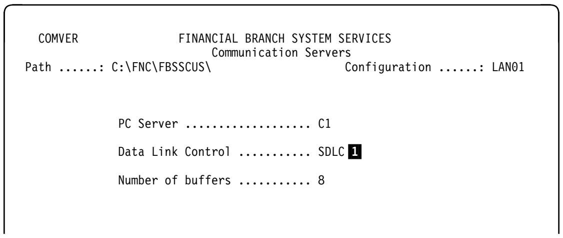
4 FBSS *N.R.Z.I.* value must match the value specified for NRZI data encoding (NRZI parameter) on the CRTLINSDLC command (see Figure G-3 on page G-2).

5 FBSS *Switched line* value must match the value specified for the connection type (CNN parameter) on the CRTLINSDLC command and the SWITCHED parameter on the CRTCTLFNC command (see Figure G-3 on page G-2).

6 Concatenation of the FBSS *Identification block* and *Identification number* must match the exchange identifier (EXCHID parameter) specified on the CRTCTLFNC command (see Figure G-3 on page G-2).

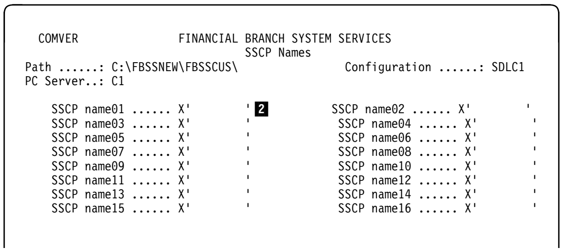
7 FBSS *Line mode* must match the value specified for the DUPLEX parameter on the CRTLINSDLC command. If *CRTS* is specified for the FBSS, DUPLEX(*FULL) must be specified on the CRTLINSDLC command (see Figure G-3).

8 Logical unit numbers specified for the FBSS controller must match the local location addresses (LOCADR parameters) specified for the device descriptions (CRTDEVFNC commands) associated with this controller description (see Figure G-4 on page G-2).



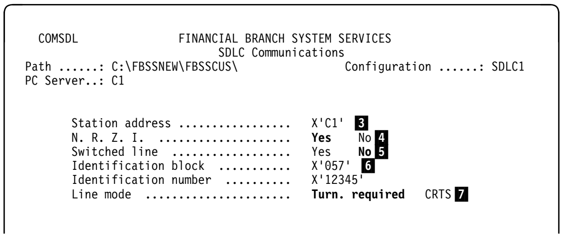
```
  COMVER                FINANCIAL BRANCH SYSTEM SERVICES
                             Communication Servers
Path ......: C:\FNC\FBSSCUS\                     Configuration ......: LAN01


              PC Server ................... C1

              Data Link Control ........... SDLC 1

              Number of buffers ........... 8
```

Figure G-1. FBSS Communication Servers Display



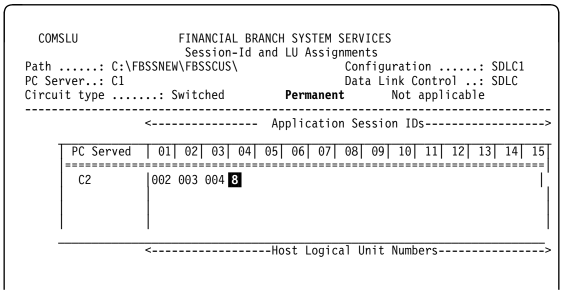
```
  COMVER               FINANCIAL BRANCH SYSTEM SERVICES
                                 SSCP Names
 Path ......: C:\FBSSNEW\FBSSCUS\                 Configuration ......: SDLC1
 PC Server..: C1

     SSCP name01 ...... X'        ' 2        SSCP name02 ...... X'        '
     SSCP name03 ...... X'        '           SSCP name04 ...... X'        '
     SSCP name05 ...... X'        '           SSCP name06 ...... X'        '
     SSCP name07 ...... X'        '           SSCP name08 ...... X'        '
     SSCP name09 ...... X'        '           SSCP name10 ...... X'        '
     SSCP name11 ...... X'        '           SSCP name12 ...... X'        '
     SSCP name13 ...... X'        '           SSCP name14 ...... X'        '
     SSCP name15 ...... X'        '           SSCP name16 ...... X'        '
```

Figure G-2. FBSS SSCP Names Display



```
  COMSDL               FINANCIAL BRANCH SYSTEM SERVICES
                             SDLC Communications
 Path ......: C:\FBSSNEW\FBSSCUS\                 Configuration ......: SDLC1
 PC Server..: C1


       Station address .................   X'C1'  3
       N. R. Z. I.  ....................   Yes    No 4
       Switched line  ..................   Yes    No 5
       Identification block  ...........   X'057'  6
       Identification number  ..........   X'12345'
       Line mode  ......................   Turn. required   CRTS 7
```

Figure G-3. FBSS SDLC Communications Display



```
  COMSLU               FINANCIAL BRANCH SYSTEM SERVICES
                          Session-Id and LU Assignments
 Path ......: C:\FBSSNEW\FBSSCUS\                 Configuration ......: SDLC1
 PC Server..: C1                                  Data Link Control ..: SDLC
 Circuit type .......: Switched        Permanent       Not applicable
 -------------------------------------------------------------------------------
                  <----------------  Application Session IDs------------------>

     | PC Served  | 01| 02| 03| 04| 05| 06| 07| 08| 09| 10| 11| 12| 13| 14| 15|
     =======================================================================|
       C2         |002 003 004 8                                            |
     |            |                                                         |
     |            |                                                         |
     _________________________________________________________________________
                  <-----------------Host Logical Unit Numbers---------------->
```

Figure G-4. FBSS Session-Id and LU Assignments Display

Configuration Example for FBSS Controller Using Token-Ring Network

The following CL commands can be used to create the AS/400 configuration objects needed to attach an FBSS controller with two attached devices to an AS/400 system over a token-ring network line.

```
CRTLINTRN  LIND(FBSSTRN) RSRCNAME(LIN031)
           ADPTADR(400010001234)
CRTCTLFNC  CTLD(CFBSS) TYPE(*FBSS) MODEL(0)
           LINKTYPE(*LAN) EXCHID(05711111)
           ADPTADR(400012345000) DSAP(04) SSAP(04)
CRTDEVFNC  DEVD(TRNDEV03) TYPE(*FNCICF) LOCADR(03)
           RMTLOCNAME(TRN03) CTL(CFBSS)
CRTDEVFNC  DEVD(TRNDEV04) TYPE(*FNCICF) LOCADR(04)
           RMTLOCNAME(TRN04) CTL(CFBSS)
```

Program Explanation

The following displays are used to configure the FBSS controller for token-ring network communications with the AS/400 system. Values that must match the AS/400 configuration are described below.

1 The *Data Link Control* specified on the FBSS Communication Servers display (*TRDLC*) must match the link type (LINKTYPE parameter) specified on the CRTCTLFNC command (see Figure G-5 on page G-3).

2 If an SSCP name is specified for the FBSS controller, the value must match the AS/400 SSCP identifier (SSCPID parameter) specified on the CRTCTLFNC command (see Figure G-6 on page G-3).

3 FBSS *Service access point for PC* value must match the destination service access point (DSAP parameter) specified on the CRTCTLFNC command (see Figure G-7 on page G-3).

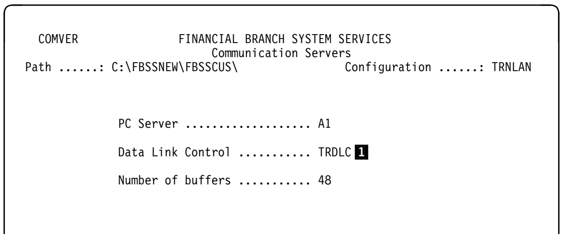
4 FBSS *Service access point for Host/37xx/4700* value must match the source service access point (SSAP parameter) specified on the CRTCTLFNC command (see Figure G-7 on page G-3).

5 FBSS *PC address* must match the value specified for the adapter address (ADPTADR parameter) on the CRTCTLFNC command (see Figure G-7 on page G-3).

6 FBSS *Host/37xx/4700 address* must match the value specified for the adapter address (ADPTADR parameter) on the CRTLINTRN command (see Figure G-7 on page G-3).

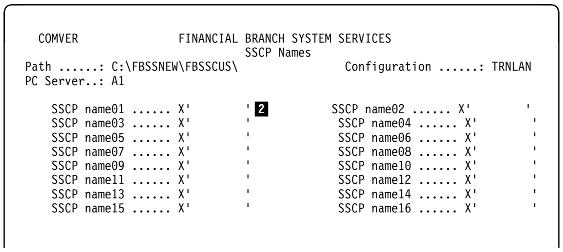
7 Concatenation of the FBSS *SNA XID Block number* and *SNA Identification number* must match the exchange identifier (EXCHID parameter) specified on the CRTCTLFNC command (see Figure G-7 on page G-3).

8 Logical unit numbers specified for the FBSS controller must match the local location addresses (LOCADR parameters) specified for the device descriptions (CRTDEVFNC commands) associated with this controller description (see Figure G-8 on page G-3).



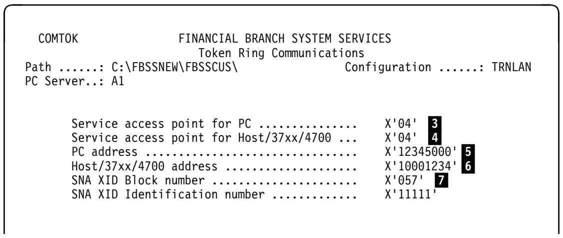
```
  COMVER              FINANCIAL BRANCH SYSTEM SERVICES
                           Communication Servers
Path ......: C:\FBSSNEW\FBSSCUS\               Configuration ......: TRNLAN


             PC Server .................. A1

             Data Link Control .......... TRDLC 1

             Number of buffers .......... 48
```

Figure G-5. FBSS Communication Servers Display



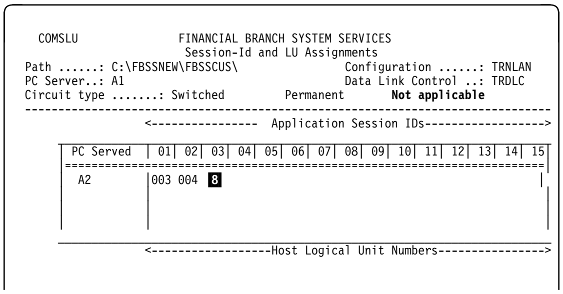
```
  COMVER              FINANCIAL BRANCH SYSTEM SERVICES
                                SSCP Names
Path ......: C:\FBSSNEW\FBSSCUS\               Configuration ......: TRNLAN
PC Server..: A1

    SSCP name01 ...... X'         ' 2         SSCP name02 ...... X'         '
    SSCP name03 ...... X'         '            SSCP name04 ...... X'         '
    SSCP name05 ...... X'         '            SSCP name06 ...... X'         '
    SSCP name07 ...... X'         '            SSCP name08 ...... X'         '
    SSCP name09 ...... X'         '            SSCP name10 ...... X'         '
    SSCP name11 ...... X'         '            SSCP name12 ...... X'         '
    SSCP name13 ...... X'         '            SSCP name14 ...... X'         '
    SSCP name15 ...... X'         '            SSCP name16 ...... X'         '
```

Figure G-6. FBSS SSCP Names Display



```
  COMTOK              FINANCIAL BRANCH SYSTEM SERVICES
                         Token Ring Communications
Path ......: C:\FBSSNEW\FBSSCUS\               Configuration ......: TRNLAN
PC Server..: A1


       Service access point for PC ...............    X'04'  3
       Service access point for Host/37xx/4700 ...    X'04'  4
       PC address ................................    X'12345000' 5
       Host/37xx/4700 address ....................    X'10001234' 6
       SNA XID Block number ......................    X'057'  7
       SNA XID Identification number .............    X'11111'
```

Figure G-7. FBSS Token-Ring Communications Display



```
  COMSLU              FINANCIAL BRANCH SYSTEM SERVICES
                          Session-Id and LU Assignments
Path ......: C:\FBSSNEW\FBSSCUS\                Configuration ......: TRNLAN
PC Server..: A1                                 Data Link Control ..: TRDLC
Circuit type .......: Switched        Permanent       Not applicable
-------------------------------------------------------------------------------
                  <----------------  Application Session IDs------------------>
         _____________________________________________________________________
        | PC Served  |  01|  02|  03|  04|  05|  06|  07|  08|  09|  10|  11|  12|  13|  14|  15|
        |=====================================================================|
        |  A2        |003 004  8                                              |
        |            |                                                         |
        |            |                                                         |
         _____________________________________________________________________
                  <------------------Host Logical Unit Numbers---------------->
```

Figure G-8. FBSS Session-Id and LU Assignments Display

Configuration Example for FBSS Controller Using X.25

The following CL commands can be used to create the AS/400 configuration objects needed to attach an FBSS controller with three attached devices to an AS/400 system over an X.25 permanent virtual circuit (PVC).

```
CRTLINX25  LIND(LFBSSX25) RSRCNAME(LIN041)
           LGLCHLE((001 *PVC)) NETADR(00000027)
           CNNINIT(*LOCAL) DFTPKTSIZE(256)
           MAXPKTSIZE(256) MODULUS(8) DFTWDWSIZE(7)
CRTCTLFNC  CTLD(CFBSSIA) TYPE(*FBSS) MODEL(0)
           LINKTYPE(*X25) SWITCHED(*NO) LINE(LFBSSX25)
           MAXFRAME(265) EXCHID(05700003) NETLVL(1984)
           LGLCHLID(001)
CRTDEVFNC  DEVD(DFBSS02) TYPE(*FNCICF) LOCADR(02)
           RMTLOCNAME(FBSS02) CTL(CFBSSIA)
CRTDEVFNC  DEVD(DFBSS03) TYPE(*FNCICF) LOCADR(03)
           RMTLOCNAME(FBSS03) CTL(CFBSSIA)
CRTDEVFNC  DEVD(DFBSS04) TYPE(*FNCICF) LOCADR(04)
           RMTLOCNAME(FBSS04) CTL(CFBSSIA)
```

To configure the FBSS controller for X.25 communications, the X.25 link profile must be customized (select *X.25 Co-Processor Link Profiles* on the Communication Profiles and Emulators display). The FBSS X.25 link profile describes the controller interface to the network; none of the values specified for the X.25 link profile need to be coordinated with AS/400 configuration values when the FBSS and AS/400 system are attached through an X.25 network.

Program Explanation

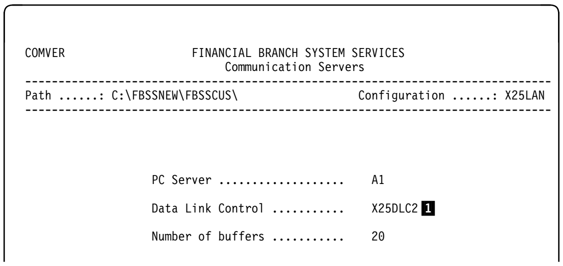
The following displays are used to complete the FBSS configuration for X.25 communications with the AS/400 system. Several of the values specified on these displays must match the AS/400 configuration. These values are described below.

1 The *Data Link Control* specified on the FBSS Communication Servers display (*X25DLC2*) must match the link type (LINKTYPE parameter) specified on the CRTCTLFNC command (see Figure G-9).

2 FBSS *Type of circuit* value must match the type of logical channel specified by the SWITCHED parameter on the CRTCTLFNC command (see Figure G-11).

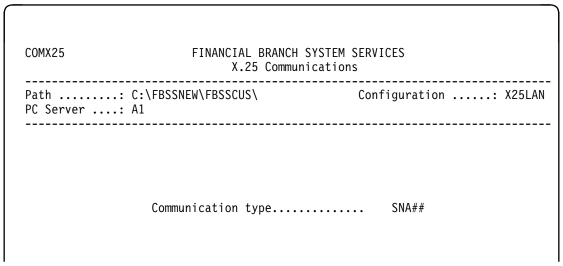
3 Concatenation of the FBSS *Identification block* and *Identification number* must match the exchange identifier (EXCHID parameter) specified on the CRTCTLFNC command (see Figure G-12).

4 Logical unit numbers specified for the FBSS controller must match the local location addresses (LOCADR parameters) specified for the device descriptions (CRTDEVFNC commands) associated with this controller description (see Figure G-13).



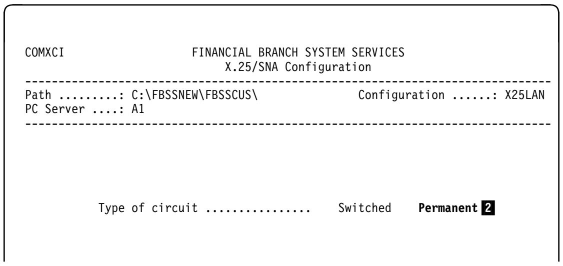
```
COMVER                  FINANCIAL BRANCH SYSTEM SERVICES
                              Communication Servers
-------------------------------------------------------------------------------
Path ......: C:\FBSSNEW\FBSSCUS\                    Configuration ......: X25LAN
-------------------------------------------------------------------------------



                  PC Server ..................    A1

                  Data Link Control ..........    X25DLC2 1

                  Number of buffers ..........    20
```

Figure G-9. FBSS Communication Servers Display



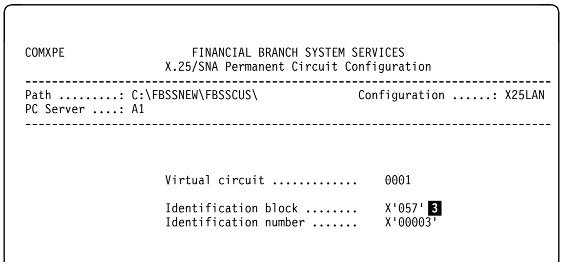
```
COMX25                  FINANCIAL BRANCH SYSTEM SERVICES
                               X.25 Communications
-------------------------------------------------------------------------------
Path .........: C:\FBSSNEW\FBSSCUS\                 Configuration ......: X25LAN
PC Server ....: A1
-------------------------------------------------------------------------------



                  Communication type.............    SNA##
```

Figure G-10. FBSS X.25 Communications Display



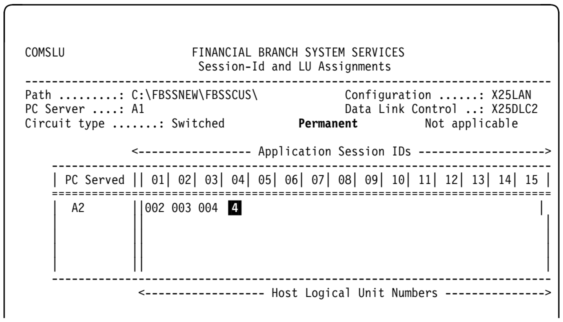
```
COMXCI                  FINANCIAL BRANCH SYSTEM SERVICES
                             X.25/SNA Configuration
-------------------------------------------------------------------------------
Path .........: C:\FBSSNEW\FBSSCUS\                 Configuration ......: X25LAN
PC Server ....: A1
-------------------------------------------------------------------------------



          Type of circuit ...............    Switched    Permanent 2
```

Figure G-11. FBSS X.25/SNA Configuration Display



```
COMXPE                  FINANCIAL BRANCH SYSTEM SERVICES
                    X.25/SNA Permanent Circuit Configuration
-------------------------------------------------------------------------------
Path .........: C:\FBSSNEW\FBSSCUS\                 Configuration ......: X25LAN
PC Server ....: A1
-------------------------------------------------------------------------------



                   Virtual circuit ............    0001

                   Identification block ........   X'057' 3
                   Identification number .......   X'00003'
```

Figure G-12. FBSS X.25/SNA Permanent Circuit Configuration Display



```
COMSLU                  FINANCIAL BRANCH SYSTEM SERVICES
                          Session-Id and LU Assignments
-------------------------------------------------------------------------------
Path .........: C:\FBSSNEW\FBSSCUS\                Configuration ......: X25LAN
PC Server ....: A1                                 Data Link Control ..: X25DLC2
Circuit type .......: Switched          Permanent          Not applicable

               <----------------- Application Session IDs ------------------->
    ---------------------------------------------------------------------------
    | PC Served || 01| 02| 03| 04| 05| 06| 07| 08| 09| 10| 11| 12| 13| 14| 15 |
    ===========================================================================
    |  A2       ||002 003 004 4                                              |
    |           ||                                                            |
    |           ||                                                            |
    |           ||                                                            |
    ---------------------------------------------------------------------------
                <------------------ Host Logical Unit Numbers --------------->
```

Figure G-13. FBSS Session-Id and LU Assignments Display

Configuration Example for 4702 Controller Using SDLC

Figure G-14 on page G-5 shows a CL program that illustrates the creation of a finance environment consisting of an AS/400 system and a 4702 finance controller with display, finance, and printer devices attached. This example corresponds to the 4702 CPGEN file that is listed after the program.

```
PGM
/********************************************************************/
/* Create the SDLC line description to go to the 4702 controller    */
/********************************************************************/
CRTLINSDLC LIND(LIN4702) RSRCNAME(LIN071) ONLINE(*NO) ROLE(*PRI) +
             NRZI(*NO)
/********************************************************************/
/* Create the 4702 controller description                           */
/********************************************************************/
CRTCTLFNC  CTLD(CTL4702) TYPE(4702) MODEL(0) LINKTYPE(*SDLC) +
             ONLINE(*NO) LINE(LIN4702) STNADR(C1)
/********************************************************************/
/* Create the device descriptions                                   */
/*        2 - 3277 display                                          */
/*        1 - 3287 printer                                          */
/*        1 - *FNCICF finance device                                */
/*        1 - 4704 finance device                                   */
/********************************************************************/
CRTDEVDSP  DEVD(FNCDSP09) DEVCLS(*RMT) TYPE(3277) MODEL(0) +
           LOCADR(09) ONLINE(*NO) CTL(CTL4702)
CRTDEVDSP  DEVD(FNCDSP10) DEVCLS(*RMT) TYPE(3277) MODEL(0) +
           LOCADR(0A) ONLINE(*NO) CTL(CTL4702)
CRTDEVFNC  DEVD(DEVICF) TYPE(*FNCICF) LOCADR(03) RMTLOCNAME(DEVICF) +
           ONLINE(*NO) CTL(CTL4702)
CRTDEVFNC  DEVD(DEV4704) TYPE(4704) LOCADR(02) ONLINE(*NO) CTL(CTL4702)
CRTDEVPRT  DEVD(FNCPRT) DEVCLS(*RMT) TYPE(3287) MODEL(0) LOCADR(08) +
           ONLINE(*NO) CTL(CTL4702)
ENDPGM
```

Figure   G-14. AS/400 Configurations Example for Finance Communications

Program Explanation

The following explains the CPGEN program shown in Figure G-15 and the specific values needed for the parameters for finance communications.

1 Since the ACB parameter is not specified, the SDLC line defaults to NRZI. Therefore, the NRZI parameter on the AS/400 SDLC line description must be specified as NRZI(*YES).

2 The TYPE parameter must be a 4502 application if an SDLC is specified.

3 The LUA parameter on the STATION macro in the CPGEN file must match the LOCADR parameter on the device description.

```
********************************************************START-ABCADM47*
***********************************************************************
**                                                                   **
**                                                                   **
**               * * * * *    4 7 0 2    * * * * *                   **
**                                                                   **
**              A B C A D M  4 7  V E R S I O N   3                  **
**                                                                   **
**                 R O C H E S T E R    C P G E N                    **
**                                                                   **
**                             F O R                                 **
**                                                                   **
**       A B C S   A N D   A D M I N   A N D   O T S                 **
**                                                                   **
**                                                                   **
***********************************************************************
***********************************************************************
**                                                                   **
** ABCS  ST02: 4704-12  77 KEYS (L1A1) ADDRESS SHARED 4710 (L1A1-4)  **
** ABCS  ST03: 4704-11  50 KEYS (L1A2) ADDRESS SHARED 4710 (L1A2-4)  **
** ADMIN ST07: 4704-22 107 KEYS (DCA1) SWAPPED FROM ABCS STATION 4   **
** ADMIN ST08: ADMIN LU1 DIRECT PRINT  USING A DPOOLED 3287 (DCA2)   **
** ADMIN ST09: 3279-2   87 KEYS (DCA3) USING A DPOOLED 3287 (DCA2)   **
** ADMIN ST10: 3279-2   87 KEYS (DCA3) SWAPPED FROM ADMIN STATION 10 **
**                                                                   **
** DEBUG = 4704-11 62 KEYS (L2A1)  UTILITY = 4704-21 62 KEYS (DCA7)  **
**                                                                   **
***********************************************************************
***********************************************************************
         EJECT
```

Figure G-15 (Part 1 of 11). CPGEN Program

4 The APBNM parameter specifies the 4702 application which runs on the specified LU address. The SNA protocol supported for the AS/400 device description at that LU address (local location address) must match the protocol supported by the application. For example, if the 4702 application supports SNA LU type 0, then the AS/400 device must be configured as a finance device.

Figure G-15 is an example of a partial CPGEN that would be used to configure a 4702 finance controller corresponding to Figure G-14 on page G-5.

```
***********************************************************************
*                                                                     *
*                  S T A R T G E N   M A C R O                        *
*                                                                     *
***********************************************************************
         SPACE
ABCADMOT STARTGEN ID=ABCADMOT,        GEN ID                           X
               DATE=042589,           GEN DATE                         X
               VN=0,                  GEN VERSION NUMBER               X
               CTLR=4702,             FOR 4702 CONTROLLER              X
               MONITOR=(EXPMB,26),    EXPANDED SYSTEM MONITOR          X
               DSKOP=N,               DISKETTE NON-OPERATIONAL         X
               TIMEOUT=2:00,          STARTUP TIME-OUT                 X
               MSGLITE=4,             MESSAGE LOG LIGHT                X
               TOALERT=Y,             TERMINAL ERROR ALERT             X
               STATS=NOWRAP,          NO WRAP STATISTICAL COUNTERS     X
               S15=65000,             GLOBAL SEGMENT- SPLIT TABLES     X
               PRINT=NOGEN
         SPACE
***********************************************************************
*                                                                     *
*                  M O N I T O R   O P T I O N S                      *
*                                                                     *
***********************************************************************
         SPACE
         MONOPTS HELPMSG=T,           INCLUDE SYSTEM MONITOR MSGS      X
               HELPCMD=T,             INCLUDE SYSTEM MONITOR HELP      X
               APMERGE=T              INCLUDE SYSTEM MONITOR MERGE
         EJECT
***********************************************************************
*                                                                     *
*                D E F I N E   T R A N S I E N T   P O O L            *
*                                                                     *
***********************************************************************
         SPACE
IPTPOOL  TRANPL  19500,RFSH=Y         DEFINE TRANSIENT STORAGE POOL
         APLIST  (IPTXFER,25,T)       INCLUDE IPTXFER AS TRANSIENT
         APLIST  (IPTXIT,25,T)        INCLUDE IPTXIT  AS TRANSIENT
*    UPDATE IPTXFER VERSION WITH EACH NEW LEVEL OF MIRCOCODE....
*    FOR MICROCODE A-4, USE VERSION 25..........................
         SPACE
***********************************************************************
*                                                                     *
*                A N D   A   U T I L I T Y   S T A T I O N            *
*                                                                     *
***********************************************************************
         SPACE
UTILITY  UTILSTAT ID=15,              UTILITY STATION IS 15 ON DCA7    X
               DA=(DCA7(0,1)),        4704 MODEL 21 USING 62 KEY KB    X
               DSKT=Y,                INCLUDE DISKETTE  FUNCTIONS      X
               DISK=Y,                INCLUDE THE DISK  FUNCTIONS      X
               DATASET=Y,             INCLUDE DATA SET  FUNCTIONS      X
               DIRECT=Y               INCLUDE DIRECTORY FUNCTIONS
         SPACE
```

Figure G-15 (Part 2 of 11). CPGEN Program

```
***********************************************************************
*                                                                     *
*        P U B L I C / S Y S T E M   V I R T U A L   V O L U M E      *
*                                                                     *
***********************************************************************
         SPACE
* VVOLUME  VIRVOL   PUBLIC=A,       PUBLIC VOLUME RESIDES ON DISK A  X
*               SYSTEM=A,           SYSTEM RESET VOLUME IS A DISK A  X
*               PVMPORT=1,          VOLUME MANAGER IS ON DCA PORT 1  X
*               VVSTA=14            USE STATION 14 FOR I/O REQUESTS
         SPACE
***********************************************************************
*                                                                     *
*        P R I V A T E   V O L U M E   F O R   D C A   P O R T   1    *
*                                                                     *
***********************************************************************
         SPACE
* VVPORT1  VVPORT   PVDRIV=A,       PRIVATE VOLUME RESIDES ON DISK A X
*               PORT=1,             PRIVATE VOLUME IS FOR DCA PORT 1 X
*           PVNAME=VVOLUME.PRIVATE1 PRIVATE VIRTUAL VOLUME DATA SET
         EJECT
***********************************************************************
*                                                                     *
*                   C O M L I N K   M A C R O                         *
*                                                                     *
***********************************************************************
         SPACE
COMLINK  COMLINK DCL=9600,           SPEED OF COMMUNICATION LINK      X
1
               TYPE=(4502), 2        COMMUNICATION TYPE IS SDLC       X
               WRAP=N,               NO WRAP CAPABILITY ON MODEM      X
               CNL=260,              SIZE   OF LINK READ BUFFERS      X
               CNB=24,               NUMBER OF LINK READ BUFFERS      X
               WRT=7,                ALLOW MAXIMUM NUMBER WRITES      X
               CTG=1,                                                 X
               OPTIONS=(BIND)
         EJECT
***********************************************************************
*                                                                     *
*                      L O O P   M A C R O S                          *
*                                                                     *
***********************************************************************
         SPACE
L1       LOOPS ID=1,                 LOOP ID                          X
               TYPE=L,               LOCAL LOOP                       X
               BPS=4800              LOOP SPEED
         SPACE
L2       LOOPS ID=2,                 LOOP ID                          X
               TYPE=L,               LOCAL LOOP                       X
               BPS=4800              LOOP SPEED
         SPACE
L3       LOOPS ID=3,                 LOOP ID                          X
               TYPE=L,               LOCAL LOOP                       X
               BPS=4800              LOOP SPEED
         SPACE
```

Figure G-15 (Part 3 of 11). CPGEN Program

```
L4       LOOPS ID=4,                  LOOP ID                          X
               TYPE=L,                LOCAL LOOP                       X
               BPS=4800               LOOP SPEED
         EJECT
************************************************************************
*                                                                      *
*                 O P T I O N A L   M O D U L E S                      *
*                                                                      *
************************************************************************
         SPACE
OPTMOD   OPTMOD                                                        X
               M45,                 FINDAP                             X
               P21,                 LTRT                               X
               P24,                 LSEEKP                             X
               P26,                 DECOMP/DECOMPTB                    X
               P27,                 COMP/COMPTB                        X
               P2A,                 SCRATCH PAD                        X
               P2C,                 INTERVAL TIMER                     X
               P42,                 DPOOL                              X
               P57,                 DES                                X
               P5E,                 SETDSKT                            X
               P68,                 LEXEC, SCALE, SETX,                X
               P70                  DATSM
         EJECT
************************************************************************
*                                                                      *
*                      F I L E S   M A C R O S                         *
*                                                                      *
************************************************************************
         SPACE
IPLDRIV  IPLDRIV  PF=1200,            PERMANENT FILE BLOCKS            X
               TF=(2,100),            TEMPORARY FILE UNITS             X
               INDX1=(1-16),          SUB FILE INDEX TF1 - ONLINE      X
               INDX2=(1-16),          SUB FILE INDEX TF2 - OFFLINE     X
               INDXC=(1-16),          COMBINED FILE INDEX - JOURNAL    X
               COMF=(1,2),            COMBINED FILE - JOURNAL          X
               LOGTM=Y,               TIME STAMP USER MESSAGES         X
               BUF=Y                  READ INDEX BUFFER
         SPACE
PRIDSKT  FILES   DKT=1,               PRIMARY DISKETTE DRIVE           X
               TF=(2,200),            TEMPORARY FILE UNITS             X
               INDX1=(1-16),          SUB FILE INDEX TF1 - ONLINE      X
               INDX2=(1-16),          SUB FILE INDEX TF2 - OFFLINE     X
               INDXC=(1-16),          COMBINED FILE INDEX - JOURNAL    X
               COMF=(1,2),            COMBINED FILE - JOURNAL          X
               EDAM=(2,2,ALLOC,XRCD,NOKEYED,MB),  INCLUDE EDAM         X
               BUF=Y                  READ INDEX BUFFER
         SPACE
AUXDSKT  FILES   DKT=2,               SECONDARY DISKETTE DRIVE         X
               TF=(2,200),            TEMPORARY FILE UNITS             X
               INDX1=(1-16),          SUB FILE INDEX TF1 - ONLINE      X
               INDX2=(1-16),          SUB FILE INDEX TF2 - OFFLINE     X
               INDXC=(1-16),          COMBINED FILE INDEX - JOURNAL    X
               COMF=(1,2),            COMBINED FILE - JOURNAL          X
               EDAM=(2,2,ALLOC,XRCD,NOKEYED,MB),  INCLUDE EDAM         X
               BUF=Y                  READ INDEX BUFFER
         SPACE
```

Figure G-15 (Part 4 of 11). CPGEN Program

```
PRIDISK  FILES   DSK=A,               PRIMARY   HARDDISK DRIVE         X
               EDAM=(2,2,ALLOC,XRCD,NOKEYED,MB)   INCLUDE EDAM
         EJECT
***********************************************************************
*                                                                     *
*                 S C R A T C H   P A D   M A C R O S                 *
*                                                                     *
***********************************************************************
         SPACE
DEFSPA1  DEFSPA ID=1,                 SCRATCH PAD ID                   X
               SIZE=3400              SCRATCH PAD SIZE
         SPACE
DEFSPA2  DEFSPA ID=2,                 SCRATCH PAD ID                   X
               SIZE=3400              SCRATCH PAD SIZE
         SPACE
DEFSPA3  DEFSPA ID=3,                 SCRATCH PAD ID                   X
               SIZE=3400              SCRATCH PAD SIZE
         SPACE
DEFSPA4  DEFSPA ID=4,                 SCRATCH PAD ID                   X
               SIZE=3400              SCRATCH PAD SIZE
         EJECT
***********************************************************************
*                                                                     *
*                   S E G S T O R   M A C R O S                       *
*                                                                     *
***********************************************************************
         SPACE
SEGSTOR1 SEGSTOR CLASS=1,             SMALL DISPLAY (TELLER ONLY)      X
               SEGSIZE=(0,            STANDARD REGISTER SEGMENT        X
               900,                   WORKSTATION ENVIRONMENT SEGMENT  X
               140,                   TELLER TRANS HOLD SEGMENT        X
               100,                   ADMIN TRANS HOLD SEGMENT         X
               100,                   TERMINAL INPUT SEGMENT           X
               650,                   TERMINAL OUTPUT & LOG SEGMENT    X
               256,                   DISKETTE INPUT/OUTPUT SEGMENT    X
               275,                   CPU INPUT/OUTPUT SEGMENT         X
               768),                  TELLER TOTALS SEGMENT            X
               S13=65000              GLOBAL SEGMENT 13
         SPACE
SEGSTOR2 SEGSTOR CLASS=2,             LARGE DISPLAY (TELLER AND ADMIN) X
               SEGSIZE=(0,            STANDARD REGISTER SEGMENT        X
               900,                   WORKSTATION ENVIRONMENT SEGMENT  X
               140,                   TELLER TRANS HOLD SEGMENT        X
               1000,                  ADMIN TRANS HOLD SEGMENT         X
               100,                   TERMINAL INPUT SEGMENT           X
               3072,                  TERMINAL OUTPUT & LOG SEGMENT    X
               256,                   DISKETTE INPUT/OUTPUT SEGMENT    X
               325,                   CPU INPUT/OUTPUT SEGMENT         X
               768,                   TELLER TOTALS                    X
               2000,                  ABCS TERMINAL SWAP TO ADMIN PP   X
               102,                   DATA DICTIONARY SCALING SEGMENT  X
               6144),                 DATA DICTIONARY ELEMENT SEGMENT  X
               S13ID=1                SHARE SEGMENT 13 WITH CLASS=1
         SPACE
```

Figure G-15 (Part 5 of 11). CPGEN Program

```
SEGSTOR3 SEGSTOR CLASS=3,             CPU COMMUNICATION WORKSTATION    X
               SEGSIZE=(200,          REGSEG - NO EXTRA BYTES          X
               360,                   WORKSTATION ENVIRONMENT SEGMENT  X
               256,                   TELLER TRANS HOLD SEGMENT        X
               70,                    ADMIN TRANS HOLD SEGMENT         X
               70,                    TERMINAL INPUT SEGMENT           X
               256,                   TERMINAL OUTPUT & LOG SEGMENT    X
               256,                   DISKETTE INPUT/OUTPUT SEGMENT    X
               325,                   CPU INPUT/OUTPUT SEGMENT         X
               0,                     NOT USED                         X
               0),                    USER SEGMENT                     X
               S13=4608
         SPACE
SEGSTOR4 SEGSTOR CLASS=4,             ADMIN PP PASSTHRU WORKSTATION    X
               SEGSIZE=(0,            REGSEG - NO EXTRA BYTES          X
               2058,                  STATION I/O BUFFER SEGMENT       X
               520,                   LOCAL STATION FIELDS SEGMENT     X
               60,                    DEFDMS DATSM MACHINE SEGMENT     X
               60,                    LOCAL ERROR AND DEBUG SEGMENT    X
               0,                     SEGMENT 6  - NOT USED            X
               0,                     SEGMENT 7  - NOT USED            X
               0,                     SEGMENT 8  - NOT USED            X
               0,                     SEGMENT 9  - NOT USED            X
               60,                    ADMIN PP TO ABCS SWAP SEGMENT    X
               0,                     SEGMENT 11 - NOT USED            X
               0),                    SEGMENT 12 - NOT USED            X
               S13=500                ADMIN PP GLOBAL FIELDS SEGMENT
         SPACE
SEGSTOR5 SEGSTOR CLASS=5,             4700 PC - FILE TRANSFER+PS/PC    X
               SEGSIZE=(0,            REGSEG - NO EXTRA BYTES          X
               3850,                  STATION I/O BUFFER SEGMENT       X
               520,                   LOCAL STATION FIELDS SEGMENT     X
               60,                    DEFDMS DATSM MACHINE SEGMENT     X
               60,                    LOCAL ERROR AND DEBUG SEGMENT    X
               0,                     SEGMENT 6  - NOT USED            X
               0,                     SEGMENT 7  - NOT USED            X
               0,                     SEGMENT 8  - NOT USED            X
               0,                     SEGMENT 9  - NOT USED            X
               60,                    ADMIN PP TO ABCS SWAP SEGMENT    X
               0,                     SEGMENT 11 - NOT USED            X
               0),                    SEGMENT 12 - NOT USED            X
               S13ID=4                ADMIN PP GLOBAL FIELDS SEGMENT
         EJECT
```

Figure G-15 (Part 6 of 11). CPGEN Program

```
***********************************************************************
*                                                                     *
*                  S T A T I O N     M A C R O S                      *
*                                                                     *
***********************************************************************
         SPACE
STATION2 STATION  ID=2,               DEFINE CONTROLLER STATION 2      X
               LUA=2, 3               TO COMMUNICATE TO HOST AS 2      X
               TERM=(L1A1),           4704-12 77 KEY, SHARED 4710      X
               DPOOL=POOL1,           POOLED 4720-4 USED PB ONLY       X
               SS=3,                  SEGMENT STORAGE IDENTIFIER       X
      4 APBNM=(ABCSAP20,1),           AP ASSOCIATED WITH THIS WS       X
               DELSET=X'02',          INITIAL DELIMITER CONTROL        X
               CPU=Y,                 HOST COMMUNICATION ALLOWED       X
               SHARED=N,              SHARED WORK STATION OPTION       X
               STARTUP=Y,             CONTROL AT START-UP OPTION       X
               WRT=1,                 LCHECK AFTER ONE LWRITE CP       X
               INTMR=0,               NO STATION INTERVAL TIMERS       X
               INSTR=65535,           ALLOW MAXIMUM BEFORE LEXIT       X
               RETSTK=12              ALLOW 12 LINK STACK LEVELS
         SPACE
STATION3 STATION  ID=3,               DEFINE CONTROLLER STATION 3      X
               LUA=3,                 TO COMMUNICATE TO HOST AS 3      X
               TERM=(L1A2),           4704-11 50 KEY, SHARED 4710      X
               DPOOL=POOL1,           POOLED 4720-4 USED JP/CF/PB      X
               SS=1,                  SEGMENT STORAGE IDENTIFIER       X
               APBNM=(ABCSAP20,1),    AP ASSOCIATED WITH THIS WS       X
               DELSET=X'01',          INITIAL DELIMITER CONTROL        X
               CPU=Y,                 HOST COMMUNICATION ALLOWED       X
               SHARED=N,              SHARED WORK STATION OPTION       X
               STARTUP=Y,             CONTROL AT START-UP OPTION       X
               WRT=1,                 LCHECK AFTER ONE LWRITE CP       X
               INTMR=0,               NO STATION INTERVAL TIMERS       X
               INSTR=65535,           ALLOW MAXIMUM BEFORE LEXIT       X
               RETSTK=12              ALLOW 12 LINK STACK LEVELS
         SPACE
         SPACE
STATION8 STATION  ID=8,               DEFINE CONTROLLER STATION 8      X
               LUA=8,                 TO COMMUNICATE TO HOST AS 8      X
               SS=4,                  SEGMENT STORAGE IDENTIFIER       X
               APBNM=(DVSAMP47,3),    AP ASSOCIATED WITH THIS WS       X
               DELSET=X'04',          INITIAL DELIMITER CONTROL        X
               CPU=Y,                 HOST COMMUNICATION ALLOWED       X
               SHARED=N,              SHARED WORK STATION OPTION       X
               STARTUP=Y,             CONTROL AT START-UP OPTION       X
               WRT=1,                 LCHECK AFTER ONE LWRITE CP       X
               INTMR=6,               ALLOW SIX  INTERVAL TIMERS       X
               INSTR=65535,           ALLOW MAXIMUM BEFORE LEXIT       X
               RETSTK=12              ALLOW 12 LINK STACK LEVELS
         SPACE
```

Figure G-15 (Part 7 of 11). CPGEN Program

```
STATION9 STATION  ID=9,               DEFINE CONTROLLER STATION 9      X
               LUA=9,                 TO COMMUNICATE TO HOST AS 9      X
               DA=(DCA3(0,1)),        3279 MODEL 2 USING 87 KEYS       X
               SS=4,                  SEGMENT STORAGE IDENTIFIER       X
               APBNM=(DVSAMP47,3),    AP ASSOCIATED WITH THIS WS       X
               DSM=(300,24,80,24,80,EAB,DCA),  DATSM - DCA + EAB       X
               DSM1=DSMNUMCK,         WITH SPECIAL NUMERIC CHARS       X
               DELSET=X'04',          INITIAL DELIMITER CONTROL        X
               CPU=Y,                 HOST COMMUNICATION ALLOWED       X
               SHARED=N,              SHARED WORK STATION OPTION       X
               STARTUP=Y,             CONTROL AT START-UP OPTION       X
               WRT=1,                 LCHECK AFTER ONE LWRITE CP       X
               INTMR=6,               ALLOW SIX  INTERVAL TIMERS       X
               INSTR=65535,           ALLOW MAXIMUM BEFORE LEXIT       X
               RETSTK=12              ALLOW 12 LINK STACK LEVELS
         SPACE
STATIONA STATION  ID=10,              DEFINE CONTROLLER STATION 10     X
               LUA=10,                TO COMMUNICATE TO HOST AS 10     X
               SS=4,                  SEGMENT STORAGE IDENTIFIER       X
               APBNM=(DVSAMP47,3),    AP ASSOCIATED WITH THIS WS       X
               DSM=(300,24,80,24,80,EAB,DCA),  DATSM - DCA + EAB       X
               DSM1=DSMNUMCK,         WITH SPECIAL NUMERIC CHARS       X
               DELSET=X'04',          INITIAL DELIMITER CONTROL        X
               CPU=Y,                 HOST COMMUNICATION ALLOWED       X
               SHARED=N,              SHARED WORK STATION OPTION       X
               STARTUP=Y,             CONTROL AT START-UP OPTION       X
               WRT=1,                 LCHECK AFTER ONE LWRITE CP       X
               INTMR=6,               ALLOW SIX  INTERVAL TIMERS       X
               INSTR=65535,           ALLOW MAXIMUM BEFORE LEXIT       X
               RETSTK=12              ALLOW 12 LINK STACK LEVELS
         SPACE
         EJECT
***********************************************************************
*                                                                     *
*           D A T S M   N U M E R I C   C H E C K   M A C R O         *
*                                                                     *
***********************************************************************
         SPACE
DSMNUMCK DSMNUMCK FC,                 ALLOW DUP   KEY  WITHIN NUMERIC  X
               FE                     ALLOW FIELD MARK WITHIN NUMERIC
         EJECT
***********************************************************************
*                                                                     *
*          D E F A D D R   A N D   D C A P O R T   M A C R O S        *
*                                                                     *
***********************************************************************
         SPACE
L1A1     DEFADDR (DS470477),          4704 MODEL 12 & 77 KEY KEYBOARD  X
               (JP471001,4),          4710 MODEL 01 - ADDRESS SHARED   X
               ADDR=(L1,1,4)          LOOP 1 ADDRESS 1 120 CPS
         SPACE
L1A2     DEFADDR (DS470450),          4704 MODEL 11 & 50 KEY KEYBOARD  X
               (JP471001,4),          4710 MODEL 01 - ADDRESS SHARED   X
               ADDR=(L1,2,4)          LOOP 1 ADDRESS 4 120 CPS
         SPACE
```

Figure G-15 (Part 8 of 11). CPGEN Program

```
L1A3     DEFADDR (JP472004,4,POOL1),  4720 MODEL 04 - DPOOLED PRINTER  X
               ADDR=(L1,3,4)          LOOP 1 ADDRESS 3 120 CPS
         SPACE
L1A4     DEFADDR (JP472004,4,POOL1),  4720 MODEL 04 - DPOOLED PRINTER  X
               ADDR=(L1,4,4)          LOOP 1 ADDRESS 4 120 CPS
         SPACE
L2A1     DEFADDR (L2A1DBUG),          4704 MODEL 11 & 62 KEY KEYBOARD  X
               ADDR=(L2,1,2)          LOOP 2 ADDRESS 1 240 CPS
         SPACE
DCA0     DCAPORT (DCA04704),          4704 MODEL 22, 107 KEY KEYBOARD  X
               PORT=0                 DCA PORT ADDRESS 0
         SPACE
DCA1     DCAPORT (DCA14704),          4704 MODEL 22, 107 KEY KEYBOARD  X
               PORT=1                 DCA PORT ADDRESS 1
         SPACE
DCA3     DCAPORT (DCA33279),          3279 MODEL 2B & 87 KEY KEYBOARD  X
               PORT=3                 DCA PORT ADDRESS 3
         SPACE
DCA7     DCAPORT (DCA7UTIL),          4704 MODEL 21 & 62 KEY KEYBOARD  X
               PORT=7                 DCA PORT ADDRESS 7
         EJECT
************************************************************************
*                                                                      *
*              D I S P L A Y   D E V X X X X   M A C R O S             *
*                                                                      *
************************************************************************
         SPACE
DS470450 DEV4704 MODEL=11,            MODEL 11 KEYBOARD DISPLAY        X
               OUTRTBL=OUT4704,       OUTPUT TRANSLATE TABLE           X
               CHARSET=US,            CHARACTER SET FOR U.S.           X
               TRTBHDR=KBABCS50,      50 KEY TRANSLATE TABLE           X
               OPTIONS=(4905,NCSR,SI),  MAGNETIC STRIPE DEVICE         X
               MSTRTBL=(MSABCSIN,MSABCSOT), MAGNETIC STRIPE TABLES     X
               ALITE=(SYS),           ALARM LIGHTS                     X
               EOMSET=X'FF',          INITIAL EOM MASK SETTING         X
               CURSOR=Y,              CURSOR TO STAY ON SCREEN         X
               ERTLS=Y                LIMIT THE KEYBOARD LREAD
         SPACE
DS470477 DEV4704 MODEL=12,            MODEL 12 KEYBOARD DISPLAY        X
               OUTRTBL=OUT4704,       OUTPUT TRANSLATE TABLE           X
               CHARSET=US,            CHARACTER SET FOR U.S.           X
               TRTBHDR=KBABCS77,      77 KEY TRANSLATE TABLE           X
               OPTIONS=(4905,NCSR),   MAGNETIC STRIPE DEVICE           X
               MSTRTBL=(MSABCSIN,MSABCSOT), MAGNETIC STRIPE TABLES     X
               ALITE=(SYS),           ALARM LIGHTS                     X
               EOMSET=X'FF',          INITIAL EOM MASK SETTING         X
               CURSOR=Y,              CURSOR TO STAY ON SCREEN         X
               ERTLS=Y                LIMIT THE KEYBOARD LREAD
         SPACE
```

Figure G-15 (Part 9 of 11). CPGEN Program

```
L2A1DBUG DEV4704 MODEL=11,            MODEL 11 KEYBOARD DISPLAY        X
               OUTRTBL=OUT4704,       OUTPUT TRANSLATE TABLE           X
               CHARSET=US,            CHARACTER SET FOR U.S.           X
               TRTBHDR=KBDBUG62,      62 KEY FOR MONITOR/DEBUG         X
               ALITE=(SYS),           ALARM LIGHTS                     X
               EOMSET=X'FF',          INITIAL EOM MASK SETTING         X
               CURSOR=Y,              CURSOR TO STAY ON SCREEN         X
               ERTLS=Y                LIMIT THE KEYBOARD LREAD
         SPACE
DCA04704 DCA4704                                                       X
               MODEL=22,              MODEL 22 IS FOR 24 X 80          X
               OUTRTBL=OUT47M2,       OUTPUT TRANSLATE TABLE           X
               TRTBHDR=KBABCS77,      107 KEY TRANSLATE TABLE          X
               TRTLIST=KBLOCL77,      107 KEY LOCAL KEY TABLE          X
               FEATURE=MSRE,          MAGSTRIPE READER/ENCODER         X
               MSTRTBL=(MSABCSIN,MSABCSOT), MAGNETIC STRIPE TABLES     X
               ALITE=(SYS),           ALARM LIGHTS                     X
               EOMSET=X'FF',          INITIAL EOM MASK SETTING         X
               CURSOR=Y,              CURSOR TO STAY ON SCREEN         X
               ERTLS=Y                LIMIT THE KEYBOARD LREAD
         SPACE
DCA14704 DCA4704                                                       X
               MODEL=22,              MODEL 22 IS FOR 24 X 80          X
               OUTRTBL=OUT47M2,       OUTPUT TRANSLATE TABLE           X
               TRTBHDR=KBABCS50,      50- KEY TRANSLATE TABLE          X
               FEATURE=MSRE,          MAGSTRIPE READER/ENCODER         X
               MSTRTBL=(MSABCSIN,MSABCSOT), MAGNETIC STRIPE TABLES     X
               ALITE=(SYS),           ALARM LIGHTS                     X
               EOMSET=X'FF',          INITIAL EOM MASK SETTING         X
               CURSOR=Y,              CURSOR TO STAY ON SCREEN         X
               ERTLS=Y                LIMIT THE KEYBOARD LREAD
         SPACE
DCA33279 DCA3279                                                       X
               MODEL=8B,              3279 MODEL 2B                    X
               OUTRTBL=OUT3278,       OUTPUT TRANSLATE TABLE           X
               TRTBHDR=KBADMN87,      87 KEY TRANSLATE TABLE           X
               ALITE=(SYS),           ALARM LIGHTS                     X
               EOMSET=X'FF',          INITIAL EOM MASK SETTING         X
               CURSOR=Y,              CURSOR TO STAY ON SCREEN         X
               ERTLS=Y                LIMIT THE KEYBOARD LREAD
         SPACE
DCA7UTIL DCA4704                                                       X
               MODEL=21,              MODEL 21 IS FOR 12 X 40          X
               OUTRTBL=OUT47M2,       OUTPUT TRANSLATE TABLE           X
               TRTBHDR=KBDBUG62,      62 KEY FOR MONITOR/DEBUG         X
               ALITE=(SYS),           ALARM LIGHTS                     X
               EOMSET=X'FF',          INITIAL EOM MASK SETTING         X
               CURSOR=Y,              CURSOR TO STAY ON SCREEN         X
               ERTLS=Y                LIMIT THE KEYBOARD LREAD
         EJECT
```

Figure G-15 (Part 10 of 11). CPGEN Program

```
************************************************************************
*                                                                      *
*              P R I N T E R   D E V X X X X   M A C R O S             *
*                                                                      *
************************************************************************
         SPACE
JP471001 DEV4710 OUTBHDR=OUT4710,     4710 OUTPUT TRANSLATE HEADER     X
               MODEL=1,               1 = JOURNAL ROLL AND CUTFORM     X
               SHARED=N,              SHARED PRINTER                   X
               KEY=N,                 AUTOSTART - START KEY NOT REQ'D  X
               WL=0,                  WARNING LINE CUTFORMS ONLY       X
               OFFSET=0,              LINES TO INDEX AT INSERT         X
               NL=AUTO,               AUTOMATIC NEW LINE FUNCTION      X
               CPI=1,                 12 CPI PRINT CHAR DENSITY        X
               LINE=48,               48 CHARS PER PRINT LINE          X
               DEFTRT=USTBL10,        4710 SPECIAL CHAR TRANSLATE TBL  X
               PS=4                   PRINT UP TO 4 LINES CUTFORM
         SPACE
JP472004 DEV4720 OUTBHDR=OUT4720,     4720 OUTPUT TRANSLATE HEADER     X
               MODEL=4,               4 = JOURNAL/CUTFORM/PASSBOOK     X
               PS=66,                 PAGE SIZE DEFAULT                X
               OFFSET=(0,00),         LINE OFFSET DEFAULT              X
               SHARED=N,              SHARED PRINTER                   X
               KEY=N,                 AUTOSTART/KEY START              X
               NL=AUTO,               NEWLINE AUTOMATIC                X
               LPI=6,                 LINES PER INCH DEFAULT           X
               CPI=1,                 CHARACTERS PER INCH DEFAULT      X
               LINE=80,               LINE LENGTH DEFAULT              X
               DEFTRT=USTBL20         4720 WORLD TABLE AT POWER ON
         SPACE
DCA24715 DEV4715 OUTBHDR=OUT4715,     4715 OUTPUT TRANSLATE HEADER     X
               LPI=5,                 LINES PER INCH DEFAULT           X
               CPI=10,                CHARACTERS PER INCH DEFAULT      X
               LINE=80                LINE LENGTH DEFAULT
         EJECT
************************************************************************
```

Figure G-15 (Part 11 of 11). CPGEN Program

Bibliography

The following AS/400 books contain information you may need. The books are listed with their full title and order number.

AS/400 Books

The following AS/400 books contain additional information you may need when you use this guide:

- *ICF Programming*, SC41-5442, contains information about writing application programs that use intersystem communications function (ICF).
- *Communications Management*, SC41-5406, provides communications support information for the AS/400 system. This includes management information, communications status and errors, and work management.
- *Communications Configuration*, SC41-5401, contains general configuration information, including detailed descriptions of network interface, line, controller, device, mode, and class-of-service descriptions, configuration lists and connection lists.
- *Remote Work Station Support*, SC41-5402, contains information and examples on how to configure your system to attach to and use remote work stations and facilities. Also, this guide contains information about the display station pass-through function.
- *DDS Reference*, SC41-5712, contains information about coding data descriptions specifications for physical, logical, display, printer, and ICF files.
- *Data Management*, SC41-5710, provides the application programmer with information about using data management support, which allows an application to work with files.
- *CL Programming*, SC41-5721, provides a wide-range discussion of AS/400 programming topics.
- *CL Reference*, contains information on control language commands.
- *Work Management*, SC41-5306, contains information on how to create an initial management environment and how to change a work management environment.
- *Security – Reference*, SC41-5302, provides information on resource security.

Programming Language Books

The following books contain information about finance communications-supported programming languages:

- *Languages: Systems Application Architecture* AD/Cycle* COBOL/400* Reference*, SC09-1380
- *Languages: Systems Application Architecture* AD/Cycle* COBOL/400* Reference Summary*, SX09-1209
- *Languages: Systems Application Architecture* AD/Cycle* COBOL/400* User's Guide*, SC09-1383
- *Languages: Systems Application Architecture* AD/Cycle* RPG/400* Reference*, SC09-1349
- *Languages: Systems Application Architecture* AD/Cycle* RPG/400* User's Guide*, SC09-1348
- *Languages: Systems Application Architecture* C/400* Reference Summary*, SX09-1217
- *Languages: Systems Application Architecture* C/400* User's Guide*, SC09-1347

Personal Banking Machine Books

The following books contain information regarding the operation and problem determination of the personal banking machines:

- *IBM 4730 Personal Banking Machine Series Customization Image Builder General Information*, GC31-0029
- *IBM 4730 Personal Banking Machine Series Network Monitor General Information*, GC31-0033
- *IBM 4731 Personal Banking Machine General Information*, GA19-5346
- *IBM 4731 Personal Banking Machine Operations Support Manual*, GA19-5378
- *IBM 4731 Personal Banking Machine Operator's Guide*, GA19-5375
- *IBM 4731 Personal Banking Machine Operator's Quick Reference*, GX11-6098
- *IBM 4731, 4732, and 4736 Personal Banking Machines Customization Guide*, GA19-5353
- *IBM 4731, 4732, and 4736 Personal Banking Machines Error Log Reference Guide*, GA19-5379
- *IBM 4732 Personal Banking Machine General Information*, GA34-2017
- *IBM 4732 Personal Banking Machine Operations Support Manual*, GA34-2020

- *IBM 4732 Personal Banking Machine Operator's Guide*, GA34-2019
- *IBM 4732 Personal Banking Machine Operator's Quick Reference*, GX31-2071
- *IBM 4736 Personal Banking Machine General Information*, GC31-0046
- *IBM 4736 Personal Banking Machine Operations Support* , GC31-0052
- *IBM 4736 Personal Banking Machine Operator's Guide*, GC31-0051
- *IBM 4736 Personal Banking Machine Operator's Quick Reference*, GC31-0053

Financial Branch System Services (FBSS) Books

The following guides contain information regarding Financial Branch System Services (FBSS):

- *IBM Financial Branch System Services Version 2.2, Application Programming*, SC19-5174
- *IBM Financial Branch System Services Version 2.2, Installation Planning and Administration Guide*, SC19-5173

Systems Network Architecture (SNA) Books

The following books contain information regarding Systems Network Architecture (SNA):

- *Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic*, SC30-3112
- *Systems Network Architecture Reference Summary*, GA27-3136

Miscellaneous Books

The following books contain additional information you may need when you use this guide:

- *Check Processing Executive/VS: Program Logic Manual*, LY20-2556
- *Check Processing Executive/VS: Program Reference and Operations*, SH20-2496
- *Check Processing Executive/3694: Program Logic Manual*, LY20-2525
- *Check Processing Executive/3694: Program Reference and Operations*, SH20-2495
- *Host Support User's Guide*, SC31-0020
- *IBM 4700 Finance Communications System Controller Programming Library*:

  Communications Programming, GC31-2068
  Control Program Generation, GC31-2071
  Cryptographic Programming, GC31-2070
  Disk and Diskette Programming, GC31-2067
  General Controller Programming, GC31-2066
  Work Station Programming, GC31-2069
- *IBM 4700 Finance Communications System: Subsystem Operating Procedures*, GC31-2032
- *IBM 4700 Finance Communications System: Subsystem Problem Determination Guide*, GC31-2033
- *IBM 4700 Finance Communications System: System Monitor Guide and Reference*, GA34-2108
- *IBM 4704 Display Station Operating Instructions*, GC31-2025

Index

Reader Comments—We'd Like to Hear from You!

AS/400 Advanced Series
Finance Communications
Programming
Version 4

Publication No. SC41-5449-00

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatis-fied	Very Dissatis-fied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

____

____

____

____

____

May we contact you to discuss your responses? __ Yes __ No
Phone: (____) ____________ Fax: (____) ____________ Internet: ____________

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937-3430**
  Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

Name ____________________ Address ____________________

Company or Organization ____________________

Phone No. ____________________

Reader Comments—We'd Like to Hear from You!
SC41-5449-00

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

ATTN DEPT 542 IDCLERK
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-9986

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC41-5449-00

Spine information:

IBM AS/400 Advanced Series

Finance Communications Programming

Version 4