Application System/400™

SC21-8099-1

Communications: Finance Communications Programmer's Guide

Application System/400™

SC21-8099-1

Communications: Finance Communications Programmer's Guide

Second Edition (September 1989)

This major revision makes obsolete SC21-8099-0.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

This edition applies to Release 2, Modification Level 0, of the IBM Operating System/400 Licensed Program (Program 5728-SS1), AS/400 Communications Utilities Program (Program 5728-CMI), and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or technical newsletters.

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.

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 licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

The numbers at the bottom right of illustrations are publishing control numbers and are not part of the technical content of this manual.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to your IBM-approved remarketer.

This publication could contain technical inaccuracies or typographical errors. A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department 245, Rochester, Minnesota, U.S.A. 55901. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

About This Guide

This guide 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.

This manual may refer to products that are announced but are not yet available.

This guide contains small programs which are furnished by IBM as simple examples to provide 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.

Who Should Use This Guide

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

What You Should Know

Before you use the material in this guide, you should be familiar with the *Programming: Control Language Programmer's Guide*, SC21-8077. This guide contains a wide-range 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.

You should also be familiar with the *Data Communications Concepts Guide*, GC21-5169. This guide provides you with an understanding of how data communications works and how to use different types of networks.

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: User's Guide*, SC21-9601, and the *Communications: Programmer's Guide*, SC21-9590.

You also need a basic understanding of finance devices and controller applications available for use on the AS/400 system. For a list of documents describing IBM finance hardware and software capabilities, see the topic "Related Printed Information" on page vi. This information helps you to configure your finance network, choose controller applications, and determine how AS/400 programs communicate with the controller applications.

If you have not yet started your 4700 system, read the *IBM 4700 Finance Communications System: Subsystem Operating Guide*, GA34-2046, for a description of how to log on and start the 4700 system.

Throughout this guide, the terms "controller" and "processor" are used interchangeably.

How This Guide Is Organized

The following list gives a brief description of the chapters and appendixes contained in this guide.

- Chapter 1 provides an introduction and overview for both ICF finance and non-ICF finance communications interfaces.
- Chapter 2 describes the support available for both ICF and non-ICF finance communications and how to establish this support on your AS/400 system.
- Chapter 3 explains how to configure both ICF and non-ICF finance communications and how to use the configuration commands for lines, controllers, and devices.
- Chapter 4 explains how to use commands to vary on and vary off communications configurations for both ICF and non-ICF finance communications interfaces.
- Chapter 5 explains how to use ICF finance functions and operations to write your programs.
- Chapter 6 explains how to use the following non-ICF command interfaces to write programs:
  - Submit Finance Job (SBMFNCJOB)
  - Finance Input/Output Manager (FIOM)
  - User-Defined Data Stream (UDDS)
- Chapter 7 discusses ICF and non-ICF finance considerations, including security, application, error recovery, and Systems Network Architecture (SNA).
- Appendix A describes the language operations, the data description specifications (DDS) keywords, and the valid system-supplied formats for ICF finance communications.
- Appendix B describes return codes, messages, and sense codes used for ICF finance communications.
- Appendix C discusses the SNA commands issued in response to ICF operations and functions issued by your program.
- Appendix D describes the diskette downloading support for non-ICF finance.
- Appendix E contains programming examples for use with ICF finance communications. These programs are written in COBOL/400 and RPG/400 languages. Source files used by these programs are also included.
- Appendix F contains programming examples for use with non-ICF finance communications. These programs are written in COBOL/400 and RPG/400 languages. Source files used by these programs are also included.

In the back of this guide is a glossary and an index. Use the glossary to find the meaning of an unfamiliar term. Use the index to look up a topic and see on which pages the topic is covered.

How This Guide Has Changed

The following major changes were made since the previous edition of this guide:

- AS/400 ICF finance support was added.
- The AS/400 finance interface presented in the Release 1.2 guide was renamed non-ICF finance. This finance information was combined with ICF finance information, whenever possible. When combining the information was not possible, the specific finance interface being discussed was indicated.
- The following appendixes were added:
  - An appendix showing language operations, supported DDS keywords, and system-supplied formats for ICF finance (Appendix A)
  - An appendix showing error return codes, sense codes, and messages for ICF finance (Appendix B)
  - An appendix showing corresponding values for ICF finance and SNA (Appendix C)
  - An appendix explaining 4701 diskette download support (Appendix D)
  - An appendix showing programming examples for ICF finance communications (Appendix E)
- The following information was deleted:
  - Command Language (CL) program example
  - One RPG example
  - Pseudocode example

  All non-ICF programming examples are in one appendix (Appendix F).

Changes made since the previous edition of this guide are indicated by vertical lines to the left of the change.

Related Online Information

The following online information is available on the AS/400 system. After pressing the Help key on any menu, you can press the Help key a second time to see an explanation of how the online information works, including the index search function. You can press either the Help key or F1 for help.

Help for Displays

You can press the Help key on any display to see information about the display. There are two types of help available:

Field
Extended

Field help explains the field on which the cursor is positioned when you press the Help key. For example, it describes the choices available for a prompt. If a system message appears at the bottom of the display, position the cursor on the message and press the Help key to see information about the cause of the message and the appropriate action to take.

Extended help explains the purpose of the display. Extended help appears if you press the Help key when the cursor is outside the areas for which field help is available.

To exit the online information, press F3 (Exit). You return to the display on which you pressed the Help key.

Index Search

Index search allows you to specify words or phrases that identify the information that you want to see. To use index search, press the Help key, then press F11 (Search index). You can also use index search by entering the Start Index Search (STRIDXSCH) command on any command line or by selecting option 2 on the User Support and Education menu.

Help for Control Language Commands

To see prompts for parameters for a control language command, type the command, then press F4 (Prompt) instead of the Enter key. To see extended help for the command, type the command on any command line and press the Help key.

Online Education

AS/400 online education provides training on a wide variety of topics. To use the online education, press F13 (User support) on any system menu to show the User Support menu. Then select the option to use online education.

Question-and-Answer Function

The question-and-answer (Q & A) function provides answers to questions you may have about using the AS/400 system. To use the Q & A function, press F13 (User support) on any system menu to show the User Support menu. Then select the option to use the question-and-answer function. You can also use the question-and-answer function by entering the Start Question and Answer (STRQST) command on any command line.

Related Printed Information

The manuals below are listed with their full title and base order number. When these manuals are referred to in this manual, a shortened version of the title is used.

The following AS/400 manuals contain additional information you may need when you use this guide:

- *Communications: Programmer's Guide*, SC21-9590, provides the application programmer with the information needed to write programs that use AS/400 communications and the intersystem communications function (ICF) file.
- *Communications: User's Guide*, SC21-9601, provides communications information that is common in the communications support for the AS/400 system. This includes setting and changing communications values, and starting and stopping communications.
- *Information Directory*, GC21-9678, identifies and describes all manuals in the AS/400 library. The manual also provides guidance on how, when, and where to use the online and printed information.
- *Programming: Control Language Programmer's Guide*, SC21-8077, provides a wide-range discussion of AS/400 programming topics.

- *Programming: Control Language Reference*, SBOF-0481, contains the commands, command parameters, syntax diagrams, default values, and keywords for the commands used in this guide.
- *Programming: Data Description Specifications Reference*, SC21-9620, provides the programmer with detailed descriptions of the entries and keywords needed to describe database files, logical and physical, and certain device files (for displays, printers, and ICF) external to the user's programs.
- *Programming: Data Management Guide*, SC21-9658, provides the application programmer with information about using data management support, which allows an application to work with files.
- *Programming: Work Management Guide*, SC21-8078, provides information about how to create an initial management environment and how to change it.

The following manuals contain information about finance communications supported programming languages:

- *Languages: C/400 Reference Summary* (available at a later date)
- *Languages: C/400 User's Guide* (available at a later date)
- *Languages: COBOL/400 Reference*, SC09-1240
- *Languages: COBOL/400 Reference Summary*, SX09-1049
- *Languages: COBOL/400 User's Guide*, SC09-1158
- *Languages: RPG/400 Reference*, SC09-1089
- *Languages: RPG/400 User's Guide*, SC09-1161

The following manuals contain additional information you may need when you use this guide:

- *Check Processing Executive/3694: Program Logic Manual*, LY20-2525
- *Check Processing Executive/3694: Program Reference and Operations*, SH20-2495
- *Check Processing Executive/VS: Program Logic Manual*, LY20-2556
- *Check Processing Executive/VS: Program Reference and Operations*, SH20-2496
- *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*, GC21-2033

- *IBM 4700 Finance Communications System: System Monitor Guide and Reference*, GA34-2108
- *IBM 4704 Display Station Operating Instructions*, GC31-2025
- *Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic*, SC30-3112
- *Systems Network Architecture Reference Summary*, GA27-3136

Contents

Figures

Chapter 1. Introduction to Finance Communications

This chapter is an introduction to AS/400[1] finance communications. An overview of the intersystem communications function (ICF) interface and the non-ICF interface, the functions these interfaces support, and an example of a finance network are also included.

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, providing a banking environment interface. Finance communications allows C/400[1], COBOL/400[1], or RPG/400[1] programs on an AS/400 system to communicate with application programs on the 4701 and 4702 finance processors and the 3694 processor. AS/400 finance communications includes two separate interfaces: intersystem communications function (ICF) finance communications and non-ICF finance communications.

Both finance interfaces communicate using the Systems Network Architecture logical unit (SNA LU) type 0 (LU0) primary protocol.

1 AS/400, C/400, COBOL/400, and RPG/400 are trademarks of the International Business Machines Corporation.

The following table provides an overview of both the ICF and non-ICF finance interfaces:

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 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 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 the interface to communicate between applications and devices

Finance Communications Network Example

Figure 1-2 is an example of an AS/400 finance communications network. This example network can be used for either ICF or non-ICF finance communications.

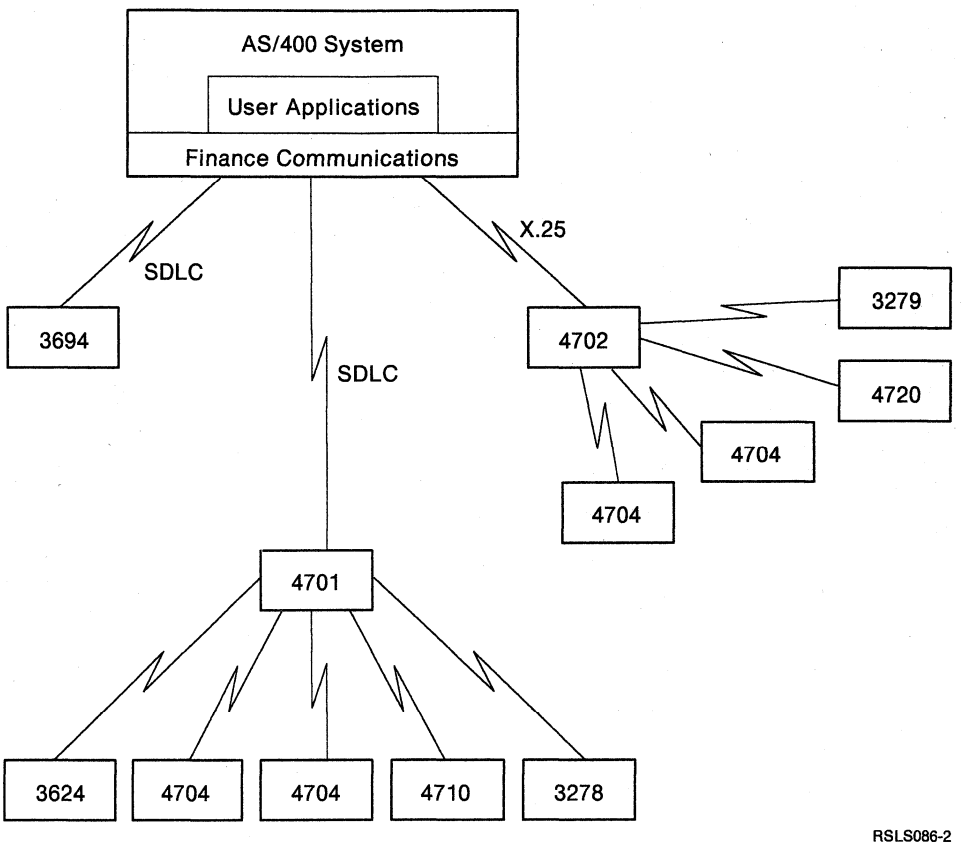


Figure 1-2. Example of an AS/400 Finance Communications Network

Chapter 2. Finance Communications Support

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

Finance Functions

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

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

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.

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 and 4702 finance controllers are programmable and perform a variety of data processing transactions in a financial 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.

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.
- Communicate with 4701 and 4702 controllers using an X.21 circuit-switching network.
- Communicate with 4701 and 4702 controllers through an X.25 packet-switching data network. The AS/400 system attaches to an X.25 network using a non-switched line through either an X.21 or X.21 bis (V.24 or V.35) physical interface.
- Allow 3270 sessions and finance sessions to share the same X.25 virtual circuit.
- Allow attached 4701 and 4702 finance controllers to share the X.25 line with any AS/400 session types possible for X.25 communications.
- Attach 3694, 4701, and 4702 finance controllers to your AS/400 system using the Synchronous Data Link Control (SDLC) protocol. The SDLC data link can be one of the following configurations for both non-ICF and ICF finance:
  - Nonswitched point-to-point
  - Nonswitched multipoint
  - Switched point-to-point, using the following connection methods:
    - Automatic dial (uses 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 3694, 4701, and 4702 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. 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 *Communications User's Guide*.

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 general system security information, see the *Security Concepts and Planning*. For finance-specific considerations, see Chapter 7, "Finance Considerations."

Error Handling and Recovery

Application programs use major and minor error return codes to handle error conditions. Applications written in C/400, COBOL/400, and RPG/400 languages can access return codes to help diagnose problems. COBOL/400 and RPG/400 programs 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 C/400 language does not support file status values. However, your 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.

You can recover from many communications errors 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 User's Guide*.

System Monitor Session

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

A system monitor session can also be defined by specifying a local location address of 01 and a finance type of *FNCICF if the ICF finance interface is used.

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 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

OS/400[1] 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:

- 3694, 4701, or 4702 controller application programs
- AS/400 application programs to process financial transactions

Application programs on the AS/400 system can be written using C/400, COBOL/400, or RPG/400 licensed programs.

Intersystem Communications Function Finance Interface

You must 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."

1 OS/400 is a trademark of the International Business Machines Corporation.

Non-Intersystem Communications Function Finance Interfaces

The non-ICF finance communications support is separate from and completely independent of ICF finance communications support. You must 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 interfaces for non-ICF finance communications:

- 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 the topic on "Using the Submit Finance Job Command" on page 6-2.

- Finance I/O Manager (FIOM) interface alone or with either the SBMFNCJOB or User-Defined Data Streams (UDDS) interface. FIOM is the recommended alternative to the UDDS interface. For more information about the FIOM interface, see the topic "Using the Finance Input/Output Manager Interface" on page 6-8.

- UDDS interface. For more information about this interface, see the topic "Using the User-Defined Data Stream Interface" on page 6-17, and the *Data Management Guide*.

Chapter 3. Configuring Finance Support

This chapter discusses line, controller, and device descriptions. You must configure these descriptions before you can begin using finance communications support.

Using Configuration Commands to Configure Finance

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 3694, 4701, or 4702 controller.

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

Command	Description
CRTLINSDLC	The Create Line Description (SDLC) command allows you to create a line description for each SDLC communications line used to communicate with the 3694, 4701, and 4702 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 and 4702 controllers.
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.
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:

Command	Description
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 interface and hex 01 through 3C for the ICF finance interface.

The device description also defines which finance interface, ICF or non-ICF, is used. If the ICF interface is used, a system monitor session can also be defined 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 ICF device type, such as 3624, 4704, and 4710.

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

Command	Description
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 the ICF finance interface, 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. Devices other than finance devices could be attached 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 60 devices for each 4701 or 4702 controller and up to four devices for each 3694 controller. Only devices as specified as TYPE(3694) or TYPE(*FNCICF) can be attached to the 3694 controller. However, any of the following device types can be configured for a 4701 or 4702 controller: 3277, 3278, 3279, 3287, 3624, 4704, and *FNCICF.

Note: If you are using the non-ICF finance interface, 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 the ICF finance interface, configure the printer as a *FNCICF type.

The device type you specify in the device description determines the format 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

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

Using 3270 Devices

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 or 4702 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.

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

You can program a 4701 or 4702 controller to do 3270 emulation. You can configure a 4704 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 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 support, see the topic about 3270 remote attachment support in the *Communications User's Guide*.

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 User's Guide*.

Using the Work with Table Commands to Configure Non-Intersystem Communications Function 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*.

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 any number of finance device tables, or to add or delete device names in existing device tables. If the device table you specify in the WRKDEVTBL command does not exist (each table must have a unique name), it is created. Device table updates can be used by any finance job that is submitted after all changes are done.

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 4701 or 4702 controller. Finance user IDs are not related to user profiles.

Use the Work with User Table (WRKUSRTBL) command to create any number of tables that define the finance user IDs that are allowed. Each table must have a unique name. If you enter a user table name that does not already exist, the table is created.

Also use the WRKUSRTBL command to add or delete finance support user IDs. User table updates can be used by any finance job that is submitted after all changes are done.

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 4701 or 4702 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 define any number of tables, but each must have a unique name. You can also use the WRKPGMTBL command to add or delete program names in these tables. If the program table you specify on the WRKPGMTBL command does not exist, it is created. Program table updates can be used by any finance job that is submitted after all changes are done.

Using the *Select Option

The Work with Device, Work with User, and Work with Program tables each have the *SELECT option available to display and select existing finance device, user, and program tables. If no tables are currently defined, the following message appears on the display:

```
NO TABLES TO DISPLAY
```

You are allowed to enter a new table name in the *New* or *Existing Table* input field.

The *SELECT option allows you to indicate the following:

SELECT a table to update
Select a table to update by entering a 2 next to the table name or by entering the table name in the *New* or *Existing Table* input field. You can only select one table at a time for updating. If you select more than one, an error message appears on the display. At the same time you select one table to update, you can also select one or more tables to be removed.

REMOVE one or more tables
Enter a 4 next to each table you want to remove. For each one you have chosen, the table name changes to *REMOVED. At the same time you select tables to be removed, you can also select one table to update, or add a new table.

ADD a new table
Add one new table at a time by entering its name in the *New* or *Existing Table* field. You can specify up to 50 characters of descriptive text for the new table on the line supplied.

CHANGE TEXT
Change text for any table by entering the new text on the line next to the table name.

DELETE TEXT
Delete text for any table by deleting the entire text field for that table.

Any text entered with the *SELECT option is carried over to the *SELECT option display.

Work with Entries Display

When you update an existing table or create a new table, the Work with Entries display appears. Extra input fields allow you to add new device, user, or program names and IDs. The program IDs are passed from the 4701 or 4702 controller during a transaction. You can have a maximum of 256 device, user, or program names in a device, user, or program table.

You can do the following operations on this display:

Add To add a device or user table, enter the new device or user name on one of the extra 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.

Enter new program IDs and program names by typing the following information on one of the extra input lines that is 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, the entry is still added to the table.

Change To change the name of any entry, enter the new information to replace the old.

Remove To remove an entry from the device, user, or program table, delete the entire entry.

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 the system monitor device on and off and starting the finance subsystem for the non-ICF finance interface are also discussed.

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 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:

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

- *OBJ: Only the specified object is varied on or off.
- *NET (default): 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.

  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.

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* manual. For more information about the VRYCFG command and WRKCFGSTS, see the *Communications User's Guide*.

Varying the System Monitor Device On and Off

If the system monitor device is varied on, you cannot vary on another device attached to the same controller. You must vary off any other device attached to the same controller before attempting to vary the system monitor device on.

Defining Communications Entries

IBM supplies two subsystem configurations to support ICF finance communications. These are QBASE, the controlling subsystem, and QCMN, the communications subsystem 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* manual.

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 User's Guide*. For more information about managing your subsystems, see the *Work Management Guide*.

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 interface.

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. The program can be coded using C/400, COBOL/400, and RPG/400 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.

Note: 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* manual.

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[1] Intersystem Communications Function (ICF) file must be created before your application can use finance communications. For more information about the ICF file, see the *Communications Programmer's Guide*.

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:

Command	Description
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.
CHGICFF	The Change ICF File command allows you to change the file attributes of the ICF file.
CRTICFF	The Create ICF File command allows you to create an ICF file specifying file level attributes.
DLTF	The Delete File command deletes the file from the system and frees the storage space allocated to that file.
DLTOVR	The Delete Override command deletes the effect of the OVRICFF command.
DLTOVRDEVE	The Delete Override Device Entry command deletes the effect of the OVRICFDEVE command.
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.
DSPOVR	The Display Override command displays overrides created by the OVRICFF file 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.

1 Operating System/400 is a trademark of the International Business Machines Corporation.

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.

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

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 *Communications Programmer's Guide*.

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. A remote location is associated with any device description that contains the same remote location name.

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 should 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 4701 or 4702 Controller

For a 4701 or 4702 controller, the ICF session is a two-part process:

1. An SNA session must be started by a request from the controller. This is considered a log on request.
2. Your program must be started, either by a request from the controller or by a manual request on the AS/400 system. This is typically considered to be the transaction, although a program which processes more than one transaction may also be started. After your program ends, the SNA session could still be active, in which case, another program may be started. This process could continue until the SNA session ends. The SNA session ends when the device is varied off (using the VRYCFG command), or when the controller requests that the session should be ended by sending a TERM-SELF command or with a Request Shutdown (RSHUTD) command.

Starting a Systems Network Architecture Session

An SNA session is started after a valid INIT-SELF is received from the controller. The AS/400 system sends a BIND command to the controller. On a secure AS/400 system, the BIND command will be sent only if the user data field of the INIT-SELF request contains a valid user ID and password. The user data field of the INIT-SELF request could also contain a default library. This 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.

For more information about the INIT-SELF command, see Figure 7-2 on page 7-9.

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: Many 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.

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.

Manually Started Program: Your program can be started by an AS/400 command or another program running on the AS/400 system. If the INIT-SELF request was not sent prior to the acquire operation issued by this program, it must be received within the time specified by the WAITFILE value of the ICF file that is used. On a secure AS/400 system, the INIT-SELF request must include a valid user ID and password. A library name specified on the INIT-SELF request is not used for manually started programs.

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:

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 7-9.

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.

If you use a manually started program, you must specify the RMTLOCNAME parameter. If the controller sent a program start request to start your program, you must specify *REQUESTER for the RMTLOCNAME parameter. For more information about starting your program, see "Starting Your Program" on page 5-4.

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.

Sending Data

Data is sent to and received from the controller by using write or 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. If you use the SDLC protocol, finance communications blocks your data into as many 256-byte records as needed. If you use the X.25 protocol with Qualified Logical Link Control (QLLC), finance communications blocks your data into as many 247-byte records as you need.

Note: Finance communications does not support multiple record groups or the X.25 protocol 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 the topic on "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 applies only to the 4701 and 4702 finance controllers. A response is not required from a 3694 processor.

Function-Management-Header Function

This function is used only with the 3694 processor.

Use the function-management-header (FMH) function to send control information to the controller program about the data following the function-management-header. You must specify data on a write operation that specifies the function-management-header. 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 4701 and 4702 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) com-

mands. 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 was 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 follows the invite function. After issuing an invite function, use the read-from-invite-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.

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). Your program can use the cancel function only if the sending of a group of records was started and was not ended. 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*.

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 *Communications Programmer's Guide.*

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. For more information about SNA considerations, see the topic on "Systems Network Architecture" on page 7-9.

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, 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 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 provide information to your program about the data record being received. When determining 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 or 02 and the minor return code is 03 for the 4701 and 4702 finance controllers. For a 3694 processor, the minor return code can also be 07 or 02.

Receive-Function-Management-Header

This function is valid only with the 3694 processor.

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 your program. 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 manual *SNA Formats*.

For more information about the input/output feedback area, see the *Communications Programmer's Guide*.

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 manual.

Note: C/400 language does not use high-level file status. However, your 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

This chapter describes the use of the following non-ICF finance interfaces:

Submit Finance Job (SBMFNCJOB) command interface
Finance Input/Output Manager (FIOM) interface
User-Defined Data Stream (UDDS) interface

Non-ICF finance communicates through a display file using UDDS. These three non-ICF interfaces represent three levels of support. The SBMFNCJOB command interface provides the most support. Your application programs are not required to do any communications operations. The FIOM interface allows your program to communicate with finance devices by calling I/O routines rather than doing communications operations. The UDDS interface gives your program total control of the data stream.

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.

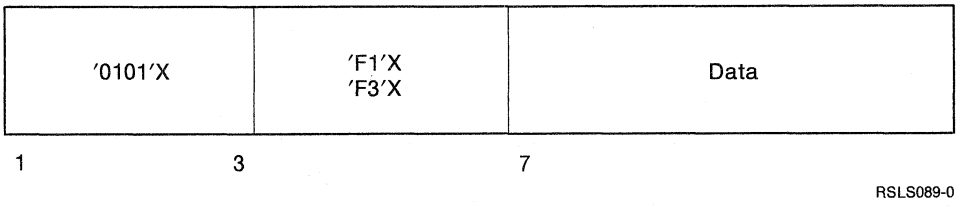


Figure 6-1. Incoming Data Control Bytes

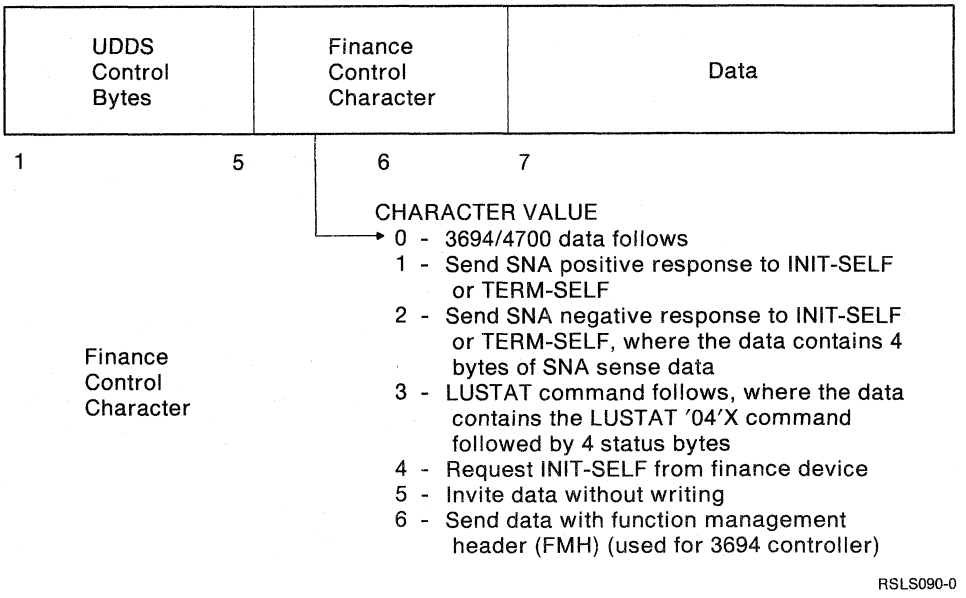


Figure 6-2. Outgoing Data Control Bytes

The following table shows the operations each interface can do without you directly specifying the operation in your program.

Figure 6-3. Interface 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 and security considerations and procedures for the non-ICF interfaces, see Chapter 7, "Finance Considerations."

Using the Submit Finance Job Command

This topic describes the Submit Finance Job (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 about the SBMFNCJOB and a syntax diagram of the command, see the *CL Reference* manual.

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 the topic "Work with Device Table Command" on page 3-4 and in the topic "Work with Program Table Command" on page 3-5.
- 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 *Communications User's Guide*.
- 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 the topic "Processing Transactions" on page 6-7.

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.

Error Handling

The SBMFNCJOB command interface gives the following error handling support.

Input/Output

Finance support attempts error recovery whenever a finance job receives an input/output (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:

```
ERRORS OCCURRED DURING FINANCE JOB INITIALIZATION OR EXECUTION

TRANSACTION FAILED

DEVICE FAILED

CONTROLLED JOB CANCELED
```

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.

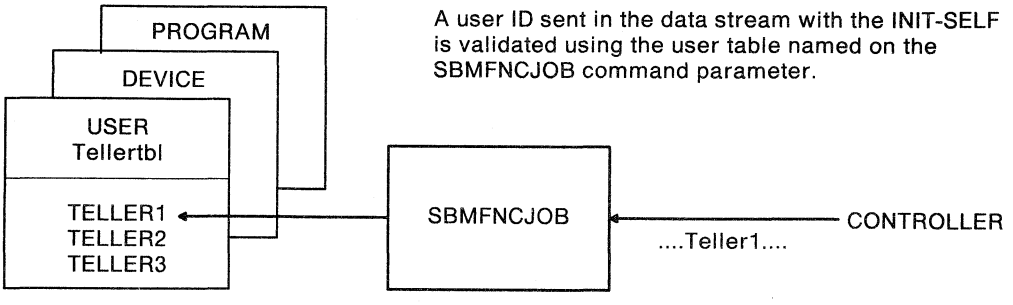
  For an explanation of the format and presentation of the IBM-supplied messages and the complete finance support message texts, see the online messages.

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

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 shows a finance job verifying that the user ID received from the controller application exists in the user table associated with that job.



SBMFNCJOB DEVTBL (DEV1) PGMTBL (PGM1) **USRTBL (TELLERTBL)** MSGQ (*WRKSTN)

RSLS052-1

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

Figure 6-5 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.

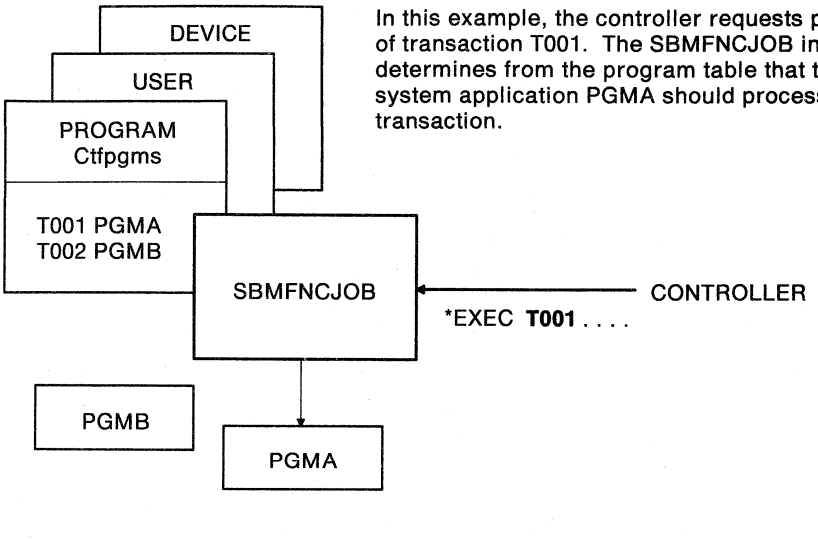


Figure 6-5. Controller Processing Transaction T001

Figure 6-6 on page 6-6 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.

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

DEVTBL1	PGMTBL1	USRTBL1
DEVICEA	PGMA PGMC PGMD	USERA : :

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

DEVTBL2	PGMTBL2	USRTBL2
DEVICEB DEVICEC	PGMA PGMB PGMD	USERB : :

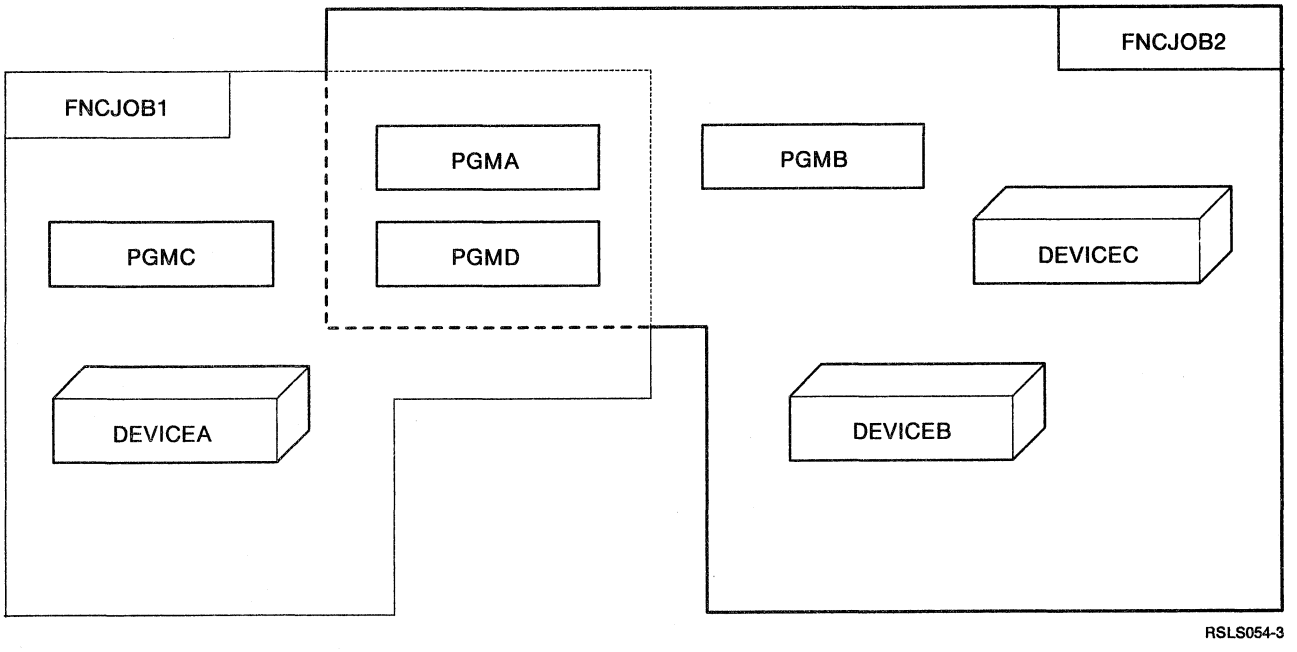


Figure 6-6. Two Finance Jobs Control the Finance Devices

Notice in Figure 6-6 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 6-6.

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 interface, the finance controller application program must use one of the data stream formats shown in Figure 6-7.

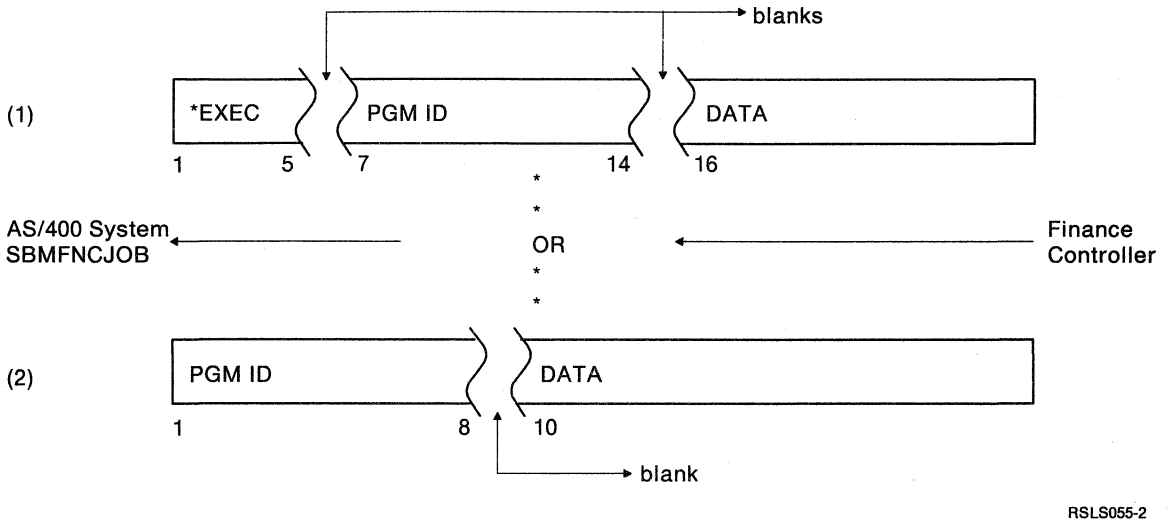


Figure 6-7. Format of Data Streams when Using the SBMFNCJOB Interface

Figure 6-7 shows two data streams being sent from the finance controller. These data streams represent the two formats accepted by the SBMFNCJOB interface. 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.

When the SBMFNCJOB interface 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 the following SBMFNCJOB parameters:

- The device name parameter: A character variable, 10 bytes in length, representing the name of the 3624 or 4704 device sending the data.
- The data length parameter: A decimal field, 15 bytes in length with 5 decimal positions, containing the length of the data received.
- The data parameter: 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.

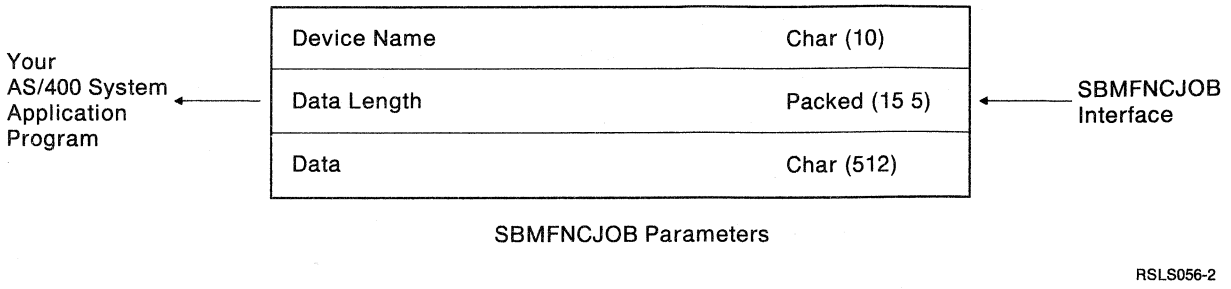


Figure 6-8. Parameters Expected by the AS/400 Application Program

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 the Finance Input/Output Manager Interface

This topic describes the Finance Input/Output Manager (FIOM) call interface, including how and when you can use it. Processing specifications and error handling are also discussed.

The FIOM interface is a flexible tool for use in a variety of finance environments. The FIOM interface supplies a high-level interface to user-defined data stream (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 3694, 4701, or 4702 controller application.

The FIOM support formats, sends, and receives UDDS. The following figures compare communications using UDDS and FIOM interfaces. 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 the UDDS control information.

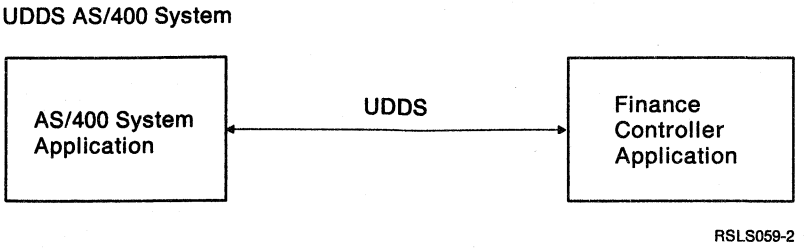


Figure 6-9. Transparent UDDS Communications

The FIOM interface makes this communication easier as shown in Figure 6-10. 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.

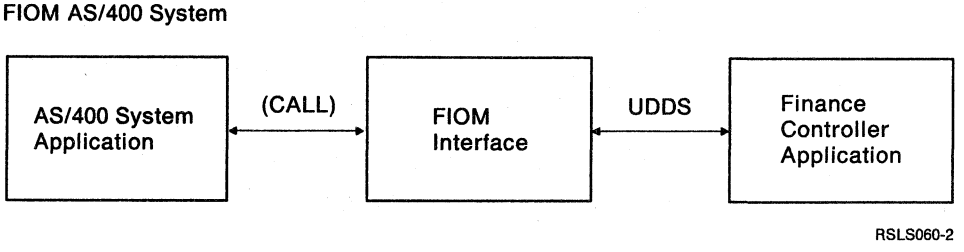


Figure 6-10. UDDS Communications with the FIOM Interface

Finance Input/Output Manager Routines

The finance input/output manager (FIOM) is an interface called by your program. The FIOM interface consists of four routines that can be called as external subroutines of your application programs. 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 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 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 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 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 shows a list of the message identifiers sent by the FIOM routines.

Figure 6-11. FIOM Error Message Table

Message ID	Message Description	QFNWRT[1]	QFNWRTI[1]	QFNREAD[1]	QFNREADI[1]
CPD8280	Device not found	X	X	X	.*no entry
CPD8281	Not authorized to device	X	X	X	.*no entry
CPD8284	Invalid data length	X	X	.*no entry	.*no entry
CPD8286	Invalid format of data length parameter	X	X	X	X
CPD8287	Invalid data type	X	X	.*no entry	.*no entry
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	.*no entry
CPF8390	Errors occurred during program processing	X	X	X	X

1 Call producing the messages.

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.

If the programming language has error-handling capabilities, the application program can attempt recovery after receiving an I/O error from the FIOM interface. 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.

For more information about the major and minor return codes used in non-ICF finance, see the *Data Management Guide*.

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

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

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

As discussed in the topic on "Processing Transactions" on page 6-7, 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 is a diagram of this communication.

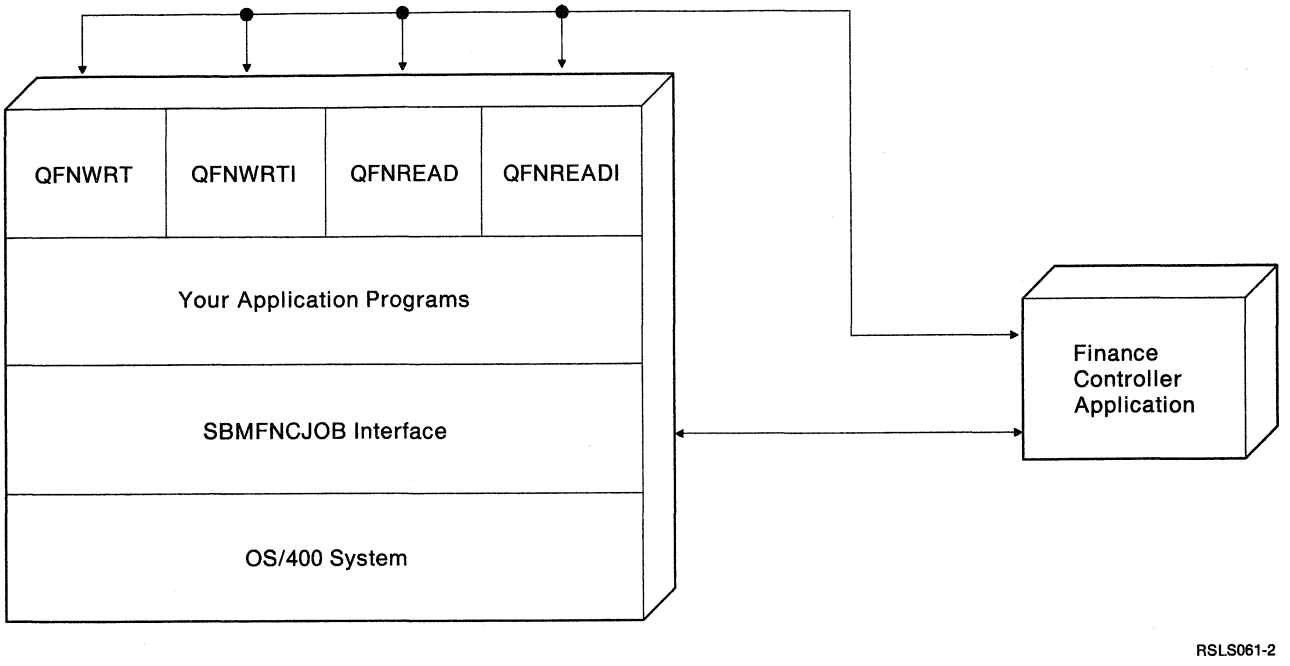


Figure 6-12. Communications between the Application Program and the Finance Controller

Figure 6-12 shows an environment in which a finance job is active on the AS/400 system using the OS/400 system. When the SBMFNCJOB interface 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 SBMFNCJOB interface, keeping the conditions previously described. The SBMFNCJOB interface 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.

An advantage gained from using the FIOM interface in a SBMFNCJOB environment is having more than 512 bytes of data associated with a particular transaction. If, for example, your 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 interface. While this capability exists through the UDDS interface, using the FIOM interface is an easier method.

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

You can use the FIOM interface 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 shows how to use the FIOM interface to handle communications, for both the communications router and the application programs.

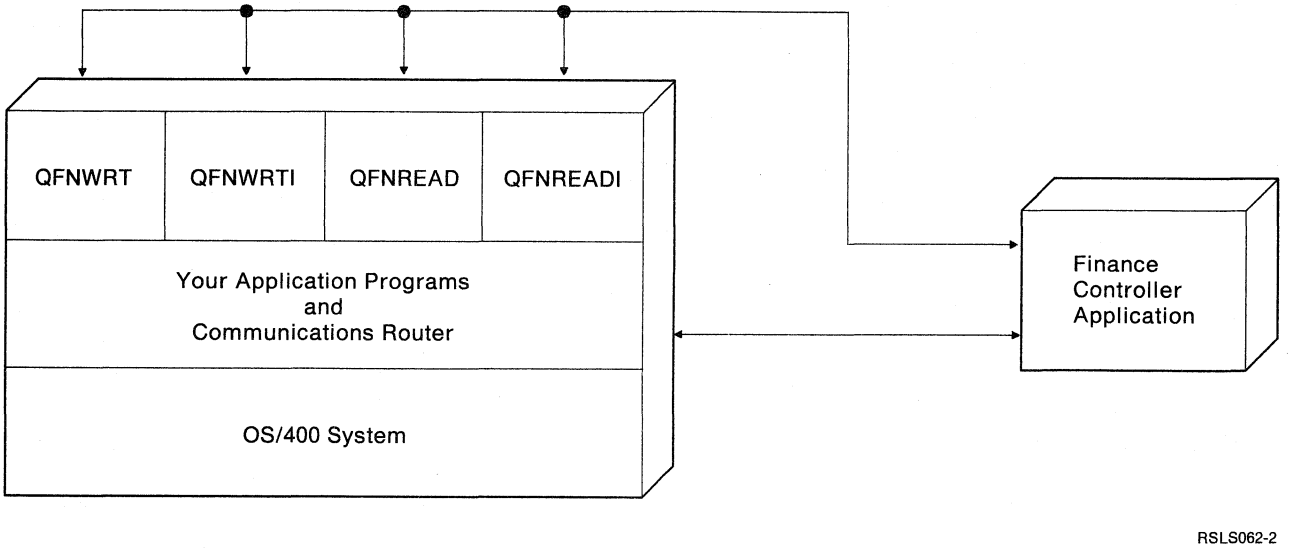


Figure 6-13. Communications for the Communications Router and Application Programs

This diagram shows an environment in which the SBMFNCJOB interface 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 the FIOM interface, a similar principle can be used to handle the commu-

nications; you can write your own subroutines to open and close the file and to acquire and release devices.

Using the User-Defined Data Stream Interface

This topic discusses formatting information for user-defined data streams (UDDS) and examples of communicating using UDDS as an interface.

Rather than using finance support, you can use the UDDS interface 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 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.

For more information on the special considerations necessary for UDDS, see the *Data Management Guide*.

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

Control Bytes

The finance control bytes for the UDDS interface are shown in Figure 6-14 and Figure 6-15.

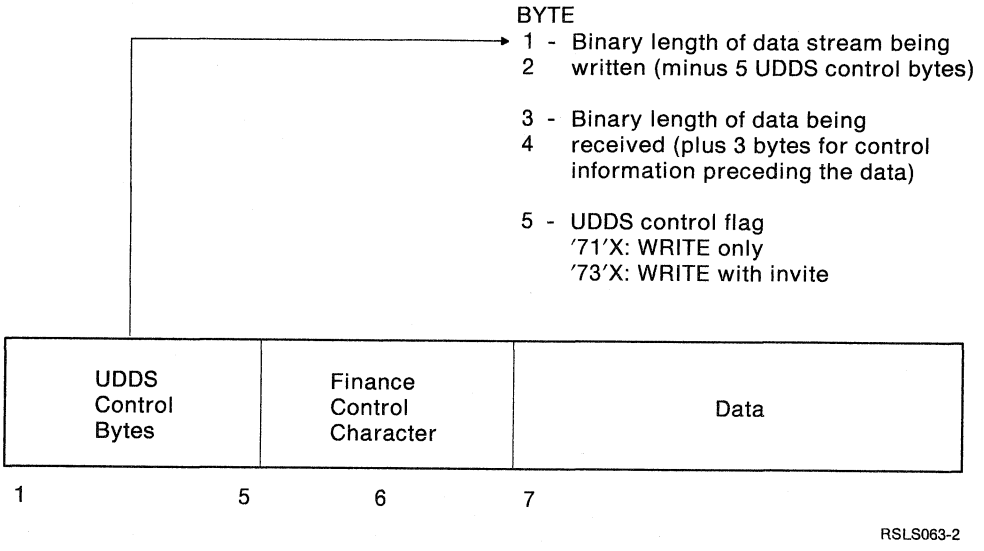


Figure 6-14. UDDS Control Bytes

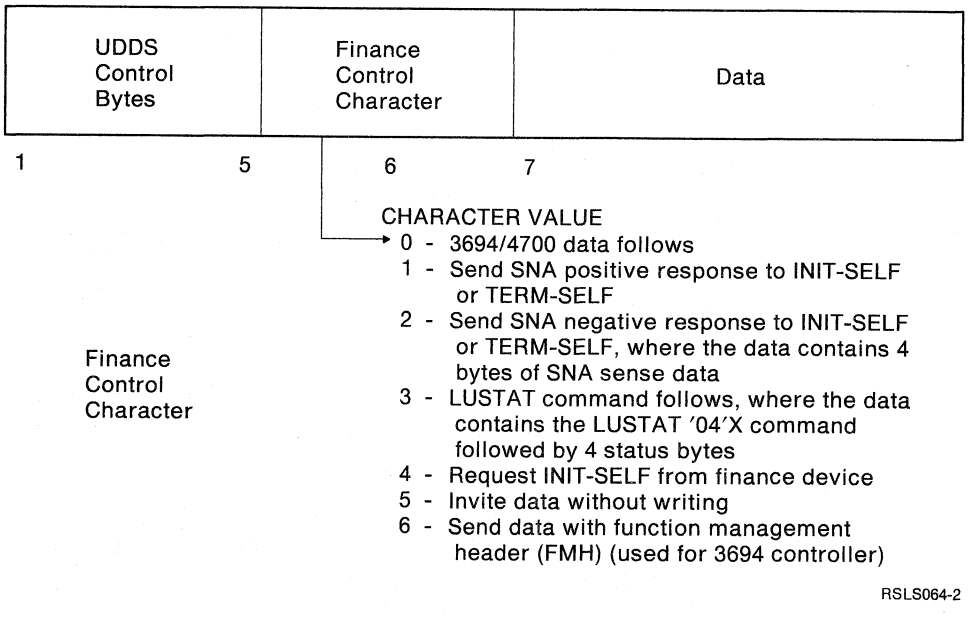


Figure 6-15. UDDS Finance Control Character

Data coming from the 3694, 4701, or 4702 controller is shown in Figure 6-16 on page 6-19. The data field follows three bytes of control information.

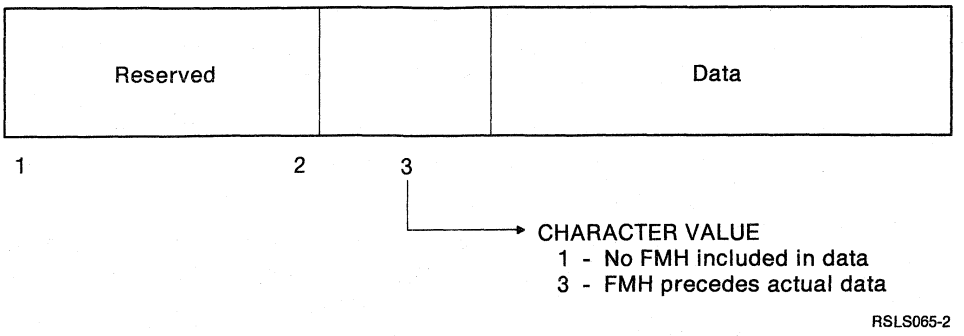


Figure 6-16. Format of Data from a 3694, 4701, or 4702 Controller

The AS/400 application program you use to communicate with a controller (using the UDDS interface) 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 the UDDS interface 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 topics describe the format of data streams that are sent and received by your AS/400 application programs when you use the UDDS interface.

INIT-SELF Request

The format for the INIT-SELF request X'0002020373F4'.

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

RSLS066-0

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'SFSbbbb' 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 the topic "Work with User Table Command" on page 3-4 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 invalid, the program may request the correct user ID.

The SBMFNCJOB interface 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
```

RSLS067-0

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
```

RSLS068-0

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 logic unit (DLU).

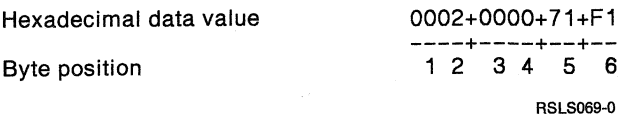
Byte 8 (X'F3')
Type of logical unit.

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'SFSbbbb' for the 4701 controller.

Positive Response to TERM-SELF Request

The format for the positive response to TERM-SELF request is X'0002000071F1'.



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.

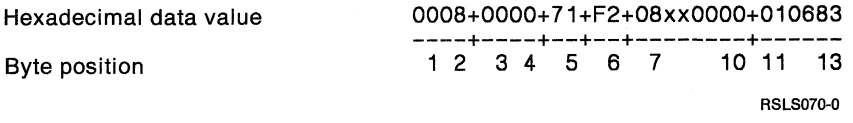
Bytes 3 and 4 (X'0000')
Length of data received from the finance controller application program. This length is set to 0 bytes to indicate that no data will be received from the controller.

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'0008000071F208xx0000010683'.



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
```

RSLS071-0

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 are 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

The SBMFNCJOB interface 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 Interface

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 the topic on "Using the User-Defined Data Stream Interface" on page 6-17)
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 the topic on "Using the User-Defined Data Stream Interface" on page 6-17)
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 the topic on "Using the User-Defined Data Stream Interface" on page 6-17)
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) read next frame 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 the UDDS interface.

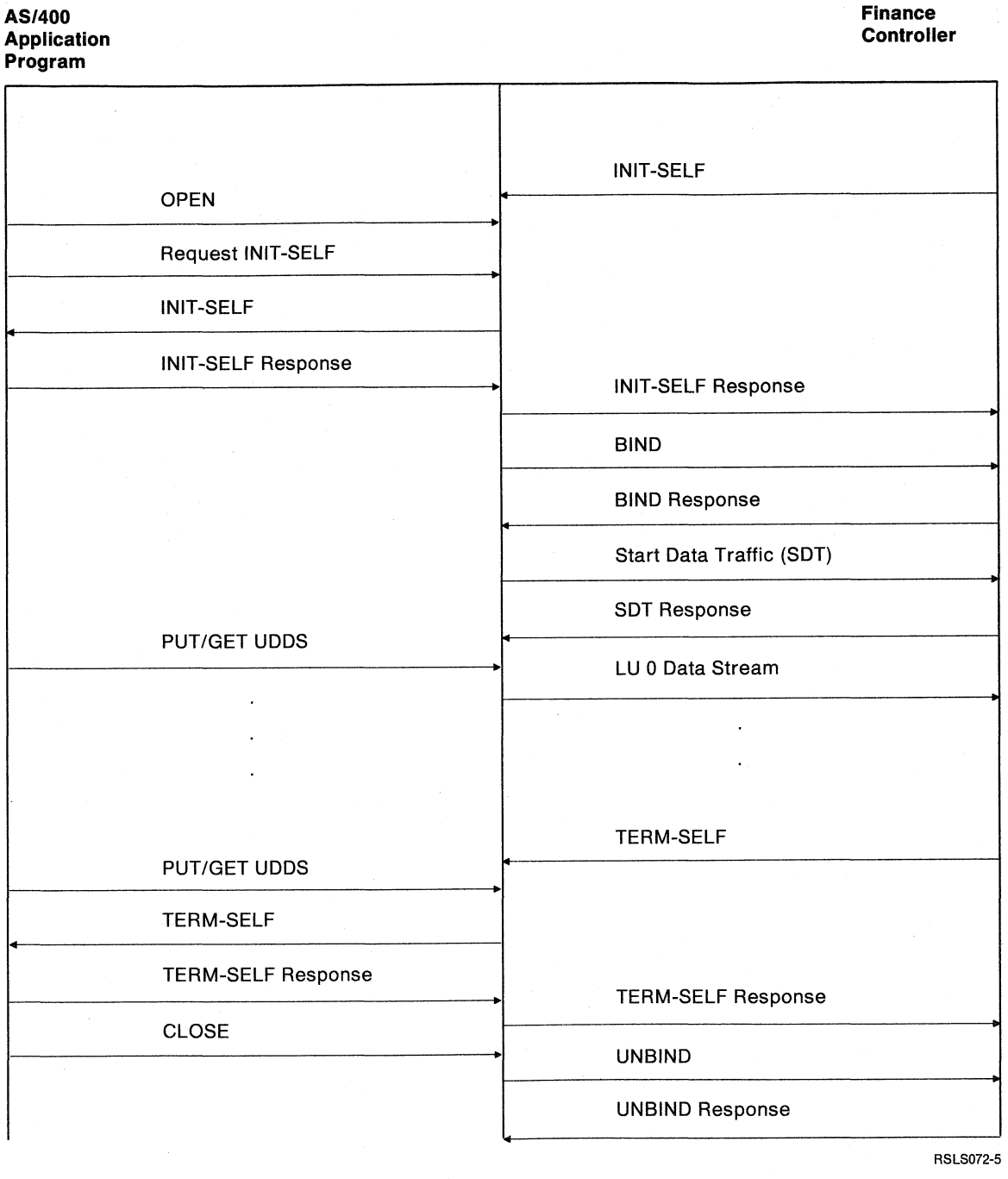


Figure 6-17. UDDS Communications Scenario

Chapter 7. Finance Considerations

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

Intersystem Communications Function

This topic 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 if 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 the topic "Using Input/Output Feedback Area" on page 5-12.
- 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 *Work Management Guide*.

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 sending of a group of records ends.

Note: 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 4701 or 4702 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 or a write-with-invite function, 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 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 *Communications Programmer's Guide*.

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 invalid parameters 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 the topic Appendix B, "Return Codes, Messages, and Sense Codes."

Non-Intersystem Communications Function

This topic discusses considerations when using non-ICF finance.

Programming Applications

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

Finance Input/Output Manager Interface

When designing AS/400 application programs, you can use the FIOM interface to do the following:

- To perform more than one write operation to the controller (rather than using the SBMFNCJOB command interface).
- 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.
- To use a display file with the same record format as the IBM-supplied display file QDFNDATA, while using the UDDS interface to communicate with finance devices.
- To 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.
- To override the QDFNDATA file by using the Override with Display File (OVRDSPF) command when defining devices to be used by the display file.
- To acquire and release finance devices used by the job, unless the FIOM interface is being used with SBMFNCJOB command.
- To 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.
- To 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.
- To 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 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.
- To 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.
- To handle any error recovery because errors received by FIOM routines result in error messages sent to your application program.

Submit Finance Job Command Interface

This command starts a continual BATCH job. The transaction programs receive controller data from and give data to the SBMFNCJOB command interface 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 interface 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:

- Determine which finance controller application program to use. This includes the following:
  - Determine the amount of work that can be unloaded from the AS/400 system to the controller. The following affects this decision:

    Amount of function supplied by the various applications at the controller level.
    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.

    Note: 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 *Work Management Guide*.

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 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 interface between the AS/400 system and the controllers, special security exists. This topic discusses these 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, see the topic on "Work with User Table Command" on page 3-4.

- Develop your AS/400 transaction application programs. Ensure that these programs do the following:

  - Accept and return the parameters for the SBMFNCJOB command interface.
  - 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 the topic on "Work with Program Table Command" on page 3-5.

- Follow the security instructions described in the topic on "Security" on page 7-8.

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, 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.

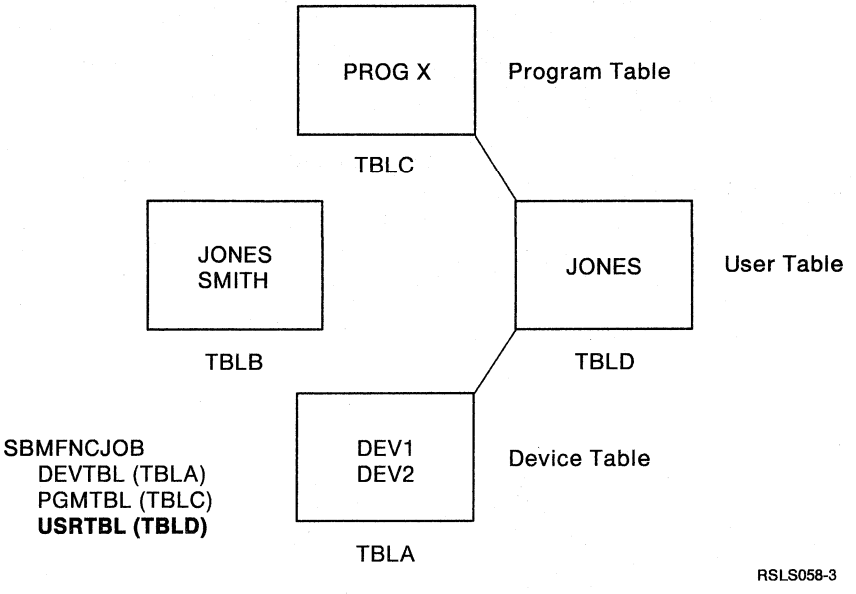


Figure 7-1. Accessing an Application Program

User-Defined Data Stream Interface

Consider the following information about the UDDS interface 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 message-queue-name parameter 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

The following topics discuss security for your finance communications network.

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 or 4702 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

Note: 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 or 4702 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

This topic discusses the SNA considerations for ICF and non-ICF finance communications. 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. The following 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 00404040404040F3
13	Length of destination logical unit name: must be hex 08
14 through 21	Not used for 4701 and 4702 controllers but must be 8 bytes in length. For the 3694 processor, this is the name of the AS/400 program to be started and must be 8 bytes in length. If the program name is less than 8 bytes, it must be 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 Intersystem Communications Function Interface

The 3694 processor follows the same procedure for initiation as for 4702 and 4702 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 the following table:

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.

4701 and 4702 Finance Controllers with Intersystem Communications Function

The INIT-SELF user data field is also used by 4701 and 4702 controllers. On a secure system, the received INIT-SELF request must contain a user ID and password but can also contain a library name. The following table shows the format of the user data field.

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

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

Finance Controllers with Non-Intersystems Communications Function Interface

For information about non-ICF, see the topic on "Sending Data from the Finance Controller to the AS/400 System" on page 6-7.

If the SBMFNCJOB interface is used, the INIT-SELF must use the format in Figure 7-2 on page 7-9 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 (BIND) command is used to establish what protocol is followed for the current session. The following table shows the BIND parameters to be used for each controller:

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

Protocol	4701 and 4702 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	0040	0040	4040

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/400, COBOL/400, and RPG/400 programming languages.

Intersystem Communications Function Operations

The following table 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. This operation is valid for only RPG/400 language.
Release	Attempts to end an ICF session.
Close	Closes the ICF file.

Intersystem Communications Function Language Statements

The following table 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	RPG/400 Operation Code	COBOL/400 Procedure Statement	C/400 Function
Open	OPEN	OPEN	fopen
Acquire	ACQ	ACQUIRE	QXXACQUIRE
Get Attributes	POST	ACCEPT	QXXDEVAT
Read	READ	READ	fread
Read-from- Invited-Program- Devices	READ	READ	QXXREADINVDEV; fread
Write	WRITE	WRITE	fwrite
Write/Read	EXFMT	Not Supported	Not Supported
Release	REL	DROP	QXXRELEASE
Close	CLOSE	CLOSE	fclose

C/400 language is case sensitive.

Data Description Specifications Keywords

Read and write operations use a record containing data description specifications (DDS) keywords. These keywords allow you to use more specific communications functions with the read and write operations. The following table 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.
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.
FMH[2]	Informs the remote program that a function-management-header is being sent.
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[2]	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.

2 Valid for *only* a 3694 processor.

System-Supplied Formats

The following table 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[2]	FMH, Invite	Write with function-management-header, then invite
$$SENDNF[2]	FMH	Write with function-management-header
$$SENDNI	No DDS keyword	Write
$$TIMER	TIMER	Set timer

1 Not valid for a 3694 Document Processor.

2 Valid for *only* a 3694 Document Processor.

Appendix B. Return Codes, Messages, and Sense Codes

This appendix includes descriptions of return codes and sense codes that are valid for ICF communications. Also included in this appendix is a table containing errors for program start requests.

Return Codes

This topic describes all the intersystem communications function (ICF) return codes that are valid for ICF 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. For more information about accessing return codes, see your high-level language manual.

Each return code is a 4-digit hexadecimal value. The first 2 digits contain the major code, and the last 2 digits contain the minor code. C/400, COBOL/400, and RPG/400 programs receive the return codes in EBCDIC hexadecimal form (4 bytes).

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

Notes:

1. In the return code descriptions, *your program* refers to the local AS/400 application program that issues the request and receives the return code from ICF communications. The *remote program* refers to the application program in the remote system with which the AS/400 application program is communicating through ICF.
2. Several references are also made in the descriptions to the operations and functions that allow your program to send and receive data or information.

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	Indication/Action
0000	**Description:** For output requests issued by your program, 0000 indicates that the last request completed successfully and that your program can continue to send data. **Action:** Issue an input or output request.
0001	**Description:** Your program successfully invited the finance session. **Action:** Issue a read-from-invited-devices request.
0003	**Description:** Your program received data and an indicator that the data is the last record in a group. Your program can continue to receive another group of records or can send data. **Action:** Issue an input or output request.
0007	**Description:** Your program received data, a function-management-header indication, and an end-of-group indication. **Action:** Issue an input or output request.

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 from a 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	Indication/Action
0203	**Description:** Your program received a group of records on a successful input request. Also, a job canceled (controlled) request is pending. **Action:** Issue an input or output request. However, the recommended action is to complete the active transaction and end the program because the system eventually cancels your job and causes all processing to stop for your job.
0207	**Description:** Your program received a function-management-header indication. Also, a job canceled (controlled) request is pending. **Action:** Issue an input or output request. However, the recommended action is to complete the active transaction and end the program because the system eventually cancels your job and causes all processing to stop for your job.

Major Code 03

Major Code 03 — Input operation completed successfully, but no data was 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	Indication/Action
0303	**Description:** Your program received a null group of records. **Action:** Issue an input or output request.
0309	**Description:** Your program is being canceled (controlled). No data was received. This return code is in response to a read-from-invited-devices operation. **Action:** Your program can continue processing. The recommended action is to complete any active transactions, release program devices, and close the file. The system eventually turns a job canceled (controlled) to a job canceled (immediate) and forces all processing to stop for your job. **Message:** CPF4741 (Notify)
0310	**Description:** The time interval specified by a timer function in your program or by the WAITRCD value specified for the ICF file ended. This return code is only applicable to the read-from-invited-program-devices operation. **Note:** Because a specific program device name is not associated with the completion of this operation, the program device name in the common I/O feedback area is set to a value of *N. Therefore, your program should not make any checks based on the program device name after receiving the 0310 return code. **Action:** Issue the operation to perform the intended function after the specified time interval ends. **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 or to cancel an invite function 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	Indication/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 already sent by the remote program. The data your program attempted to send was not sent and should be sent later, after the data or response indication from the remote program is received. **Action:** Issue an input request to receive the data or response indication. **Messages:** CPF5076 (Notify) CPF4750 (Notify)

Major Code 08 and Major Code 11

Major Codes 08-11 — 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 return code descriptions for the appropriate recovery actions.

Code	Indication/Action
0800	**Description:** The acquire operation was not successful. Your program tried to acquire a session that already was acquired and is still active. **Action:** If the session requested by the original acquire operation is the one needed, your program can begin communicating in the session because it is already available. If you want a different session, issue another acquire operation for a different session by specifying a different program device name. (The program device name must be specified in the PGMDEV parameter of the ADDICFDEVE or the OVRICFDEVE command that precedes the program.) **Messages:** CPD4077 (Diagnostic) CPF50A0 (Status)
1100	**Description:** The read-from-invited-program-devices operation was not successful because your program attempted this operation before an invite function was issued. **Action:** Issue an invite function followed by a read-from-invited-program-devices operation. **Message:** 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 the input buffer of the program or was not compatible with the record format specified on the input operation.

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

Code	Indication/Action
3401	**Description:** The input buffer used was less than the length of the received data. **Action:** Issue another input operation if your program can specify a record size large enough to receive the actual data length plus any indicators for files without separate indicator areas. Otherwise, end the session, close the file, correct the record size, and run the program. **Message:** CPF4768 (Notify)
3441	**Description:** A valid record format name was specified and the format selection type is *RECID. Although the data received matches one of the record formats in the file, it does not match the format specified on the read operation. **Action:** Correct the program to issue an input 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. **Message:** CPF5058 (Notify)
3451	**Description:** The file record size specified when the file was opened is not large enough for the data and indicators received (for files defined without a separate indicator area). For files that do not use a separate 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. **Message:** CPF4768 (Notify)

Major Code 80

Major Code 80 – Permanent system or file error (not recoverable).

Description: A file or system error that is not recoverable 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. Recovering 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 return code description.

- Close the file, open the file again, then establish the session. If the operation still is 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 from the point where the session error occurred.

Code	Indication/Action

8081 **Description:** The operation attempted was unsuccessful because a system error condition was detected.

Action: Your communications configurations need to be varied off and then on again. Your program can (1) continue local processing; (2) close the file, open the file again, and acquire the session again; or (3) end.

Messages:

```
CPF5197 (Escape)
CPF5244 (Escape)
CPF5250 (Escape)
CPF5257 (Escape)
CPF5274 (Escape)
CPF5355 (Escape)
CPF5346 (Escape)
CPF5347 (Escape)
```

8082 **Description:** The operation was not successful because the device for the remote location is not usable. For example, communications stopped for the device by an immediate Hold Communications Device (HLDCMNDEV) command, or a cancel reply was given to an error recovery message for the device. No operations should be issued to the device.

Action: Communications with the remote program cannot continue until the device is reset to a varied on state. If the device was 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 continue local processing, or it can end.

Message:

```
CPF5269 (Escape)
```

80B3 **Description:** The open operation was unsuccessful because the file was not available.

Action: The file cannot be opened until the necessary resources are available.

Message:

CPF4128 (Escape)

80EB **Description:** An open operation was tried, but an open option value that was not valid was specified. The open operation may have failed because (1) a value of update or delete is used to open the file, but the value is not supported by the device, or (2) there is a mismatch between whether or not a separate indicator area is used between your program and the ICF file.

Action: Close the file. When the problem is corrected, open the file again.

If the open option specified is incorrect, close the file and issue the open operation again with a correct open option for the device you are trying to open.

If there is a mismatch in the specification of the separate indicator, either change the ICF file (INDARA keyword) or the program.

If there is level checking, either recompile the program to match the file level of the ICF file, or change or override the ICF file by specifying LVLCHK as *NO on the Change ICF File (CHGICFF) or Override ICF File (OVRICFF) commands.

Messages:

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

80ED **Description:** An open operation was tried but an open value that was not valid was specified. The open operation failed because there is a file level check between your program and the ICF file.

Action: Close the file. Either recompile the program to match the file level of the ICF file, or change or override the ICF file by specifying LVLCHK as *NO on the Change ICF File (CHGICFF) or the Override ICF File (OVRICFF) commands.

Message:

CPF4131 (Escape)

80EF **Description:** You attempted an open operation to a file for which you are not authorized.

Action: Close the file, correct the problem, then issue the open operation again. For authority errors, obtain authority to the device from your security officer or device owner. If the device is in service mode, dedicated service tools (DST) is currently using the device. Wait until the device is available before you issue the operation again.

Message:

CPF4104 (Escape)

80F8 **Description:** An open operation to a file was not successful because the file is already open, or it is in error.

Action: If the file is already open, close the file and end the program. Remove the duplicate open operation from your program, and issue the open operation again.

Messages:

CPF4132 (Escape)
CPF5129 (Escape)
CPF5293 (Escape)

Major Code 81

Major Code 81 — Permanent system error (not recoverable).

Description: A system error that was not recoverable 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 be expected to work.

Action: You can perform the following general actions for all 81xx return codes. Specific actions are given in each return code description.

If your program started the session, you can:

- Correct the problem, and establish the session again. If the operation is not successful, your program should end the session.
- Continue processing without the session.
- End.

If your session was initiated by a program start operation from the remote program, you can:

- Continue processing without the session.
- End.

Several 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, and 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 return 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 from the point where the session error occurred.

Code	Indication/Action
8140	**Description:** A cancel reply was received from the operator, program, or system default value for a previous inquiry or notify message. The session is no longer active. **Action:** If your program initiated the session, issue the acquire operation to start the session again. If your program was started by a received program start request, it can continue local processing or end. **Message:** CPF5104 (Escape)
8191	**Description:** A permanent line error occurred on an input or output operation, and the system operator took a recovery option in response to the line error message. (You can find out what type of line error occurred by asking the system operator.) The session ended. **Action:** If your program started the session, issue the acquire operation to start the session again. If your program was started by a received program start request, it can continue local processing or end. **Messages:** 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 ended. **Action:** If your program started the session, use another acquire operation to start the session again. If your program was started by a received program start request, it can continue local processing or end. **Messages:** CPF5167 (Escape) CPF5241 (Escape)
81A3	**Description:** The SNA session was ended abnormally due to 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 and try the request again. **Message:** CPF5167 (Escape)
81A4	**Description:** SNA protocol violation occurred. 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 and try the request again. For more information about sense data, see the topic "Sense Codes" on page B-29 and the *Systems Network Architecture Formats* manual. **Messages:** CPF5167 (Escape) CPF5248 (Escape)

81AD **Description:** An attempt to establish an SNA session was not successful. The Synchronous Data Link (SDLC) frame size is not large enough to contain the request/response unit (RU) size. This was either a configuration error, or the SDLC frame size was negotiated to be a smaller value by the AS/400 system and the remote controller via the Exchange ID (XID) command.

Action: A value of 256 bytes is used for the RU size. The frame size is specified on the MAXFRAME parameter of the line description. Verify that the configuration parameter is correct. End the program. If configuration changes must be made, vary the device off first, and then vary the device on again. Try the request again.

Messages:

CPF4260 (Escape)
CPF5341 (Escape)

81BA **Description:** A data record was received that exceeds the maximum user record length.

Action: End the program and increase the size of the maximum record length (MAXRCDLEN parameter) on the ICF file if necessary. Increase the size of the input buffer on the record format to be used for the input operation. Then try the transaction again. Verify that the data sent from the remote program was correct.

Message:

CPF5205 (Escape)

81E9 **Description:** The format selection value for the file is *RECID and an input operation was issued, but the data received does not match any record formats in the file. If all formats in the file are defined with an *RECID keyword, no default record formats can be used.

Action: Verify that the data sent by the remote program was correct. Change the program to send the correct data. If the received data was correct, add an *RECID keyword definition to the file that matches the data received, or define a record format in the file without a *RECID keyword so that a default record format can be used on input operations.

Message:

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 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 operation from a remote session, 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 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, and then vary the configuration on.
- To change a parameter value for 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 (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	Indication/Action
8209	**Description:** Your open or acquire operation was not successful because the prestart job was canceled. This can be caused by one of the following: • An End Job (ENDJOB), End Prestart Job (ENDPJ), End Subsystem (ENDSBS), End System (ENDSYS), or Power Down System (PWRDWNSYS) command was issued. • The maximum number of prestart jobs (MAXJOBS) parameter value was reduced by the Change Prestart Job Entry (CHGPJE) command. • The maximum number of program start requests (MAXUSE) parameter value was exceeded. • Too many unused prestart jobs exist. • The prestart job has an error in the initialization. **Action:** Determine the cause of the problem, correct the error and start this job again. **Messages:** CPF4292 (Escape) CPF5313 (Escape)
8221	**Description:** An SNA command that was received for a remote location or device description was either not supported or was not valid. 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 and try the request again. For more information about sense data, see the topic "Sense Codes" on page B-29 and the *Systems Network Architecture Formats* manual. **Message:** CPF5206 (Escape)
8233	**Description:** A program device name that was not valid was detected. Either an ADDICFDEVE or OVRICFDEVE command was not done, or the program device name in your program does not match the program device name specified in the ADDICFDEVE or the OVRICFDEVE command for the session being acquired. The session was not started. **Action:** If the error is in your program, specify the correct program device name in your program. If an incorrect identifier was specified on the ADDICFDEVE or the 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. The file was previously in error, or the file could not be opened due to a system error. **Action:** Your communications configurations need to be varied off and then on again. Your program can (1) continue local processing; (2) close the file, open the file again, and acquire the program device again; or (3) end.

Messages:

CPF4143 (Escape)
CPF4168 (Escape)
CPF4265 (Escape)
CPF4231 (Escape)
CPF4254 (Escape)
CPF4304 (Escape)
CPF5197 (Escape)
CPF5202 (Escape)
CPF5244 (Escape)
CPF5250 (Escape)
CPF5257 (Escape)
CPF5274 (Escape)
CPF5355 (Escape)

8282 **Description:** The open or acquire operation was not successful because the device for the remote location is not usable. For example, communications were stopped for the device by a Hold Communications Device (HLDCMNDEV) command, or a cancel reply was given to an error recovery message for the device. No operations should be issued to the device. No session was started.

Action: Close the file. Communications with the remote program cannot continue until the device is reset to a varied on state. If the device was 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. When the device is reset, communications can start again by opening the file again and acquiring the device.

Messages:

CPF4298 (Escape)
CPF5269 (Escape)

8291 **Description:** A permanent line or station error occurred on an open or acquire operation, and the system operator took a recovery option in response to the error message. (Ask the system operator to find out what type of error occurred). The session ended.

Action: If your program started the session, issue the acquire operation to start the session again. If your program was started by a received program start request, it can continue local processing or end.

Messages:

CPF4261 (Escape)
CPF4193 (Escape)
CPF5260 (Escape)
CPF5324 (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: If the communications session was established, end the session. Otherwise try the request again when the remote controller is available.

Messages:

CPF4178 (Escape)
CPF5241 (Escape)

82A2 **Description:** An SNA INIT-SELF request was received for a finance remote location or device description that did not contain valid authorization data. One of the following occurred:

- User ID or password was not supplied
- User ID was not found on the system
- Password was not valid for this user ID
- User ID was not authorized to use this device description

Action: Correct the user ID or password specified 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.

Messages:

CPF4177 (Diagnostic)
CPF5251 (Escape)

82A4 **Description:** An SNA protocol violation occurred. 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 and try the request again. For more information about sense data, see the topic "Sense Codes" on page B-29 and the *Systems Network Architecture Formats* manual.

Messages:

CPF4141 (Escape)
CPF5248 (Escape)

82A6 **Description:** During an open operation, one of the following occurred:

- A negative response with sense data was received when the the BIND or Send Data Traffic (SDT) command was sent to the user to start the session.
- The BIND SDT command did not end in the time limit specified using the wait time (WAITFILE) parameter on the ADDICFF, CHGICFF, or OVRICFF command.

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. For more information about sense data, the *Systems Network Architecture Formats* manual.

Messages:

CPF4142 (Escape)
CPF5240 (Escape)

82A7 **Description:** The acquire operation was not successful. The device description specified for a program device is already being used by another program device in the same file.

Action: Omit the open operation.

Messages:

CPF4106 (Escape)
CPF4254 (Escape)
CPF5202 (Escape)
CPF5507 (Escape)

82A8 **Description:** The acquire operation was not successful because the maximum number of program devices allowed for the ICF file was 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, then close your file and increase the MAXPGMDEV value in the ICF file.

Messages:

CPF4254 (Escape)
CPF4745 (Diagnostic)
CPF5041 (Status)
CPF5202 (Status)

82A9 **Description:** The acquire operation was not successful because the *REQUESTER device was not available or the *REQUESTER device was already acquired.

The *REQUESTER device may not be available because:

- The job does not have a *REQUESTER device; that is, the job was not a batch job that was started by a program start request.
- The job was started by a program start request with the *REQUESTER device detached.
- The *REQUESTER device was released because an end-of-session was requested.
- A permanent error occurred on the session.

Action: Your program can continue local processing, or it can end.

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 permanent error codes. It is the responsibility of the remote program to initiate error recovery.

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 device. Your program is attempting to use two program devices that specify RMTLOCNAME(*REQUESTER) in the corresponding ICF device entry.

Messages:

CPF4366 (Escape)
CPF5380 (Escape)
CPF5381 (Escape)

82AA **Description:** The acquire operation was not successful because a remote location is not configured on the system that matches the remote location definition specified on the ADDICFDEVE or OVRICFDEVE command. The remote location definition is determined by the RMTLOCNAME parameter on the device description.

Action: Your program can continue local processing or close the file and end. Verify that the name of the remote location (with which your program is attempting to communicate) is specified correctly in the ADDICFDEVE or OVRICFDEVE command.

Messages:

CPF4363 (Escape)
CPF4364 (Escape)
CPF5378 (Escape)
CPF5379 (Escape)

82AB **Description:** The acquire operation was not successful because the device description for the remote location is 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.

Messages:

CPF4285 (Escape)
CPF5333 (Escape)

82AD **Description:** An attempt to establish an SNA session was not completed successfully. The SDLC frame size was not large enough to contain the RU size. This was either a configuration error, or the SDLC frame size was negotiated to be a smaller value by the AS/400 system an remote controller via the XID command.

Action: A value of 256 is used for the RU size. The frame size is specified on the MAXFRAME parameter of the line description. Verify that the parameter is correct, then end the program. If configuration changes must be made, vary the device off first and then vary the device on again. Try the request again.

Message:

CPF5341 (Escape)

82B3 **Description:** The acquire operation was not successful because all of the sessions specified in the communications type configuration are already in use. Therefore, the session was not started.

Action: Wait for one of the sessions in the communications type to become available; then send the acquire operation again. Otherwise, continue local processing or end.

Messages:

CPF4282 (Escape)
CPF5332 (Escape)

82EA **Description:** An open or acquire operation was not successful. A record format selection of *RECID was specified on the open operation, but it cannot be used on the file because the *RECID DDS keyword is not used on any of the record formats in the file.

Action: Close the file. Change the record format selection parameter (FMTSLT) to select formats by means other than *RECID, or use a file that has *RECID DDS keywords specified for at least one record format. Open the file again.

Messages:

CPF4348 (Escape)
CPF5521 (Escape)

82EC **Description:** The acquire operation was not successful because finance communications does not support FMTSLT(*RMTFMT).

Action: End your program, correct either the ADDICFDEVE or OVRICFDEVE command, and then start your program again.

Messages:

CPF4347 (Escape)
CPF5515 (Escape)

82EE **Description:** An open or acquire operation was attempted to a device that is not supported.

Your program is trying to acquire a device that is not a valid communications type, or it is trying to acquire the requesting program device in a program that was not started because of a received program start request.

Action: Your program can continue local processing, or close the file and end.

Verify that the name of the remote location (with which your program is attempting to communicate) was specified correctly on the ADDICFDEVE or OVRICFDEVE command.

If your program was attempting to acquire a requesting program device, verify that your program is running in the correct environment.

Messages:

CPF4223 (Escape)
CPF4251 (Escape)
CPF4760 (Escape)
CPF5550 (Escape)

82EF **Description:** An open or acquire operation was attempted to a device that is in service mode or you are not authorized to the device.

Action: For an open operation, close the file, correct the problem, then issue the open operation again. For an acquire operation, correct the problem and issue the acquire operation again. Dedicated service tools (DST) is currently using the device. Wait until the device is available before you issue the operation again. For authorization errors, obtain authority for the device from your security officer or device owner.

Messages:

CPF4104 (Escape)
CPF4186 (Escape)
CPF5278 (Escape)
CPF5279 (Escape)

82F4 **Description:** The acquire operation was not successful because the open operation for input only is only valid for a requesting program device.

Action: End your program, correct either the ADDICFDEVE or OVRICFDEVE command, and then start your program again.

Messages:

CPF4322 (Escape)
CPF5539 (Escape)

Major Code 83

Major Code 83 — Session error occurred (the error is recoverable).

Description: An error occurred during an I/O operation, but the session is still 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 controller was not active at the time, a second attempt could 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 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, and then vary the configuration on.
- To change a parameter value for the file, use the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.

Note: When a parameter can be specified in both 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	Indication/Action
830B	**Description:** Your program attempted an input or output operation either before the session was acquired or after it ended. The session ended by a release operation or an end-of-session function being issued, or by a permanent error. Your program may have incorrectly handled a permanent session or session not acquired error. **Action:** Verify that your program tries no input or output operation without an active session and that the return code is handled correctly. If you want your program to recover from an incorrectly handled error condition, issue another acquire operation.

Messages:

CPF4079 (Diagnostic)
CPF5067 (Escape)
CPF5068 (Escape)
CPF5070 (Escape)

8319 **Description:** A negative-response function with sense data was issued by the other program.

Action: Your program should examine the sense data in the device-dependent part of the I/O feedback area to determine the necessary error recovery. For more information about sense data, see the topic "Sense Codes" on page B-29 and the *Systems Network Architecture Formats* manual.

Messages:

CPF4813 (Notify)
CPF4814 (Notify)

831B **Description:** Sense data that was not valid was specified on a negative-response function. The NEGRSP was not sent.

Action: Your program must send sense data with the following format for the first 4 bytes: 10xx, 08xx, or 0000. Your program can also issue a negative-response function without sense data, in which case, finance communications sends the code 08110000 to the other program. If your program specifies sense data, it must be 8 bytes in length.

Message:

CPF4820 (Notify)

831C **Description:** Your program received an output exception (return code 0412) on a previous output operation and it issued another output operation.

Action: Issue an input operation to receive the data or response indication, and correct the error in your program so that it does not do an output operation when a return code of 0412 is received.

Message:

CPF4934 (Notify)

831E **Description:** The operation just issued by your program was not valid or a combination of operations that were not valid was specified. The session is still active. The error was caused by one of the following:

- Either your program issued an operation that was not supported, or the operation specified a DDS keyword is not supported by this ICF communications type.
- An output request with the invite function was issued for a file that was opened for output only.
- A close operation was issued 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 or issue a release operation, end-of-session function, or end. Correct the error in your program before attempting to communicate with the other program.

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, release the session, close the file, and open the file again for input and output.

Messages:

CPF4790 (Notify)
CPF5149 (Escape)

831F **Description:** The length of the data record or the data specified on the operation was not valid.

On an output operation, your program may have tried to send a data record having a length that exceeds the maximum user record length specified for the session, or the program may have issued a read or write operation that specified a data length greater than the record format in the device file. The session is still active.

If the function was a timer function, the format of the timer interval was not HHMMSS, where HH is hours, MM is minutes, and SS is seconds. If the operation was another system-defined format or VARLEN, the length of the user buffer was not valid.

Action: Your program can issue an output operation again with a smaller output length or you can change the record length in your program and compile it again.

For an input operation, specify a length equal to or less than the record format record length, or do not specify a length on the read operation.

If the send operation used the VARLEN keyword, verify that the length specified is less than the length specified for the file when it was opened. If the send operation used the timer function, verify that the format of the timer interval is HHMMSS.

Messages:

CPF4762 (Notify)
CPF4765 (Notify)
CPF4767 (Notify)

8322 **Description:** A fail or negative-response function was issued when your program was sending data, or when no data was received or sent that could have failed. The negative-response function is not valid when your program is in send state. The fail function is not valid when your program is not in the process of sending a group of records or data is not received for which an error can be sent. The session is still active.

Action: Your program can issue an output operation to continue sending, issue an input operation to begin receiving, issue an end-of-session function, or end. Correct the error in your program before attempting to communicate with another program.

Messages:

CPF4822 (Notify)
CPF4823 (Notify)

8323 **Description:** A cancel function was issued when data was received for your program.

Action: Your program can issue an input operation to continue receiving,

issue an end-of-session function, or end. Correct the error in your program before attempting to communicate with another program.

Message:

CPF4776 (Notify)

8326 **Description:** A cancel function was issued to cancel a group of records when no records were previously sent to start a group.

Action: Correct your program so it does not attempt to cancel a group of records when no records are sent to start a group.

Message:

CPF4779 (Notify)

832C **Description:** A release operation that was not valid, following an invite function, was detected in your program. Because your program issued the invite function, it cannot issue a release operation to end the invited session.

Action: Issue a read or read-from-invited-program-devices operation to complete the invite function or write 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 in your program caused the error, correct your program.

Message:

CPF4769 (Notify)

832D **Description:** An operation that was not valid, following an invite function, was detected in your program. For the same session, when you issue an invite function, another invite function, a cancel function, or a negative-response function, these functions cannot be issued until the first invite function is completed by a read or read-from-invited-program-devices operation, or is canceled with a cancel-invite function.

Action: Issue an input operation to receive the data that was possibly received before you issue another invite function or cancel the invite with the cancel-invite function. Otherwise, issue an end-of-session function to end the session. If a coding error in your program caused the error, correct your program.

Message:

CPF4924 (Notify)

832F **Description:** A release operation was completed unsuccessfully. A release operation is not valid if a group of records was partially sent or received, or if any data or response indicators were received from the remote program for which your program did not issue an input operation.

Action: If your program started sending a group of records, it should finish sending the group. Otherwise, your program should issue an input operation to receive the data or response indicator received from the remote program. Your program can end the transaction and session by issuing an end-of-session operation.

Messages:

CPF4818 (Notify)
CPF4819 (Notify)

83B6 **Description:** The remote program has quiesced the SNA session on which this transaction is running. This was done by sending an SNA quiesce-at-end-of-chain (QEC) command. The remote program may release the quiesced state at a later time by sending an SNA release quiesce (RELQ) command.

Action: Your program may wait and try the output operation at a later time. The controller program may have released the quiesced state. Or, your program may continue and receive data from the controller program, detach the transaction, end the session, or end the job.

Message:

CPF4816 (Notify)

83E0 **Description:** Your program attempted to issue an operation using a record format that is not defined for the file.

Action: Examine the name of the record format in your program to be sure it is correct, then verify that the record format is defined in the file definition.

Message:

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 function that was not valid.

Message:

CPF4763 (Notify)

83F8 **Description:** Your program tried an operation on a program device that was previously marked in error. Your program may have handled the error incorrectly.

Action: Release the program device and then acquire it again before attempting any more I/O operations to it.

Message:

CPF5293 (Escape)

Program Start Request Errors

When a program start request is rejected by the system, message CPF1269 is sent to the system operator message queue. This message contains information that can be used to determine why the program start request was rejected.

The SNA sense code shown in the following table is sent to the remote system issuing the program start request. The reason code is specified in the associated message text of message CPF1269.

Figure B-1 (Page 1 of 3). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description
401	Attach request received for a device that is not allocated to an active subsystem
403	User profile cannot be accessed
404	Job description cannot be accessed
405	Output queue cannot be accessed
406	Maximum number of jobs, defined by subsystem description, is already active
407	Maximum number of jobs, defined by communications entry, is already active
408	Maximum number of jobs, defined by routing entry, is already active
409	Library on library list is in use exclusively by another job
410	Group profile cannot be accessed
411	Insufficient storage in machine pool to start job
501	Job description is not found
502	Output queue is not found
503	Class is not found
504	Library on library list is not found
505	Job description or job description library is damaged
506	Library on library list is destroyed
507	Duplicate libraries are found on library list
508	Defined size of storage pool is zero
602	Value of transaction program name is reserved but not supported
604	Matching routing entry is not found
605	Program is not found
704	Password is not valid
705	User is not authorized to device
706	User is not authorized to subsystem description
707	User is not authorized to job description
708	User is not authorized to output queue
709	User is not authorized to program
710	User is not authorized to class
711	User is not authorized to library on library list

Figure B-1 (Page 2 of 3). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description
712	User is not authorized to group profile
713	User ID is not valid
723	There is no password associated with the user ID
801	Program initialization parameters are present but not allowed
802	More than 2000 bytes of program initialization parameters received for the prestart job
803	Subsystem ending in progress
804	Prestart job is either not active or is ending
805	WAIT(*NO) specified on prestart job entry
806	MAXJOBS on prestart job entry exceeded
807	Prestart job ended too soon
901	Program initialization parameters are not valid
902	Number of parameters for program is not valid
903	Program initialization parameters required but not sent
1001	System logic error; function check or unexpected return code encountered
1002	System logic error; function check or unexpected return code encountered while receiving initialization parameters
1501	Character in procedure name not valid
1502	Procedure not found
1503	System/36 environment library not found
1504	Library QSSP not found
1505	File QS36PRC not found in library QSSP
1506	Procedure name is greater than 8 characters
1507	Current library not found
1508	Not authorized to current library
1509	Not authorized to file QS36PRC in current library
1510	Not authorized to procedure in current library
1511	Not authorized to System/36 environment library
1512	Not authorized to file QS36PRC in System/36 environment library
1513	Not authorized to procedure in System/36 environment library
1514	Not authorized to library QSSP
1515	Not authorized to file QS36PRC in library QSSP
1516	Not authorized to procedure in file QS36PRC in library QSSP
1517	Unexpected return code from System/36 environment support
1518	Problem phase program not found in library QSSP
1519	Not authorized to problem phase program in library QSSP
1520	Maximum number of target programs started (100 per System/36 environment)

Figure B-1 (Page 3 of 3). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description
2651	*EXEC statement not specified
2652	Blank missing after *EXEC statement
2653	Program name missing
2654	Program name greater than 10 characters
2655	Library name greater than 10 characters

Sense Codes

This topic provides a list of valid sense codes sent by ICF communications to finance controllers for various errors.

Request Reject Error (Category Code = X'08')

This category indicates that the request was delivered and was understood and supported, but not run.

Category and description (in hexadecimal numbers):

080F End user not authorized: The requesting end user does not have access to the system or the requested resource.

0815 Function active: An INIT-SELF was received when the LU-LU session was already active.

0816 Function inactive: A TERM-SELF was received when no LU-LU session was active.

081A Request sequence error: The sequence of requests is not valid.

081C Request cannot be processed: The requested function cannot be processed due to an error condition in the receiver.

Bytes 2 and 3 following the sense code contain sense code-specific information. Allowed settings are:

0002 The receiver has an error resulting from a software problem that prevents processing the request.

0826 Function-management-header function not supported: An unsupported SSCP function was requested. Only INIT-SELF and TERM-SELF requests are supported.

0828 Reply not allowed: A request requires a normal-flow reply, but the outbound data flow for this session is quiesced, and there is no delayed reply capability.

0835 Invalid Parameter: An INIT-SELF was received that contained a field that was not valid.

Bytes 2 and 3 contain a two-byte binary count that indexes (zero-origin) the first byte of the fixed- or variable-length field having contents that are not valid (invalid fields in the INIT-SELF request).

084B Requested resources not available: Resources named in the request are not currently available. It is not known when the resources will be available. It is possible that the user profile cannot be accessed, or the device is held as a result of the HLDCMNDEV command.

084C Permanent insufficient resource: The receiver cannot act on the request because resources required to honor the request are permanently not available. The sender should not try the request again because the situation is not temporary.

0864 Function canceled: The conversation ended abnormally. A system error occurred.

Request Error (Category Code = X'10')

This category indicates that the request/response unit (RU) was delivered to the intended network addressable unit (NAU), but could not be interpreted or processed. This condition represents a mismatch of NAU capabilities.

Category and description (in hexadecimal numbers):

1001 RU data error: Data in the request RU is not acceptable. The program name specified in the user data field of the INIT-SELF command is not valid.

1002 RU length error: The request RU was too long or too short or a specified resource name (program, library) was too long.

1003 Function not supported: The function requested is not supported. The function may have been specified by a formatted request code, a field in an RU, or a control character.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 A session control (SC) command was received.

0003 An unsupported normal-flow data flow control (DFC) command was received.

0004 An unsupported expedited-flow DFC command was received.

1006 Required field or parameter is missing.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0002 A required program name was omitted on a program start request. A required program name was omitted on a program start request. A program name was not supplied following the *EXEC.

State Error (Category Code = X'20')

This category indicates a sequence number error or a request header (RH) or request/response unit (RU) that is not allowed for the receiver's current session control or data flow state. These errors prevent delivery of the request.

Category and description (in hexadecimal numbers):

2001 Sequence number: Sequence number received on normal-flow request was not one greater than the last.

2002 Chaining: Error in the sequence of the chain indicator settings (begin chain indicator (BCI) or end chain indicator (ECI)), such as first, middle, first indicator.

Bytes 2 and 3 following the sense code contain specific information on sense codes. Settings allowed are:

0001 A middle or end-of-chain request was received while in the between-chain state.

0002 A begin chain request was received while in the in-chain state.

200A Immediate request mode error: The immediate request mode protocol has been violated by the request.

Bytes 2 and 3 following the sense code contain specific information on sense codes. Settings allowed are:

0000 No specific code applies.

0001 The receiver received a middle or end-chain request when in the between-chain state.

0002 The receiver received a begin-chain request when in the in-chain state.

Request/Response Header Usage Error (Category Code = X'40')

This category indicates that the value of a field or a combination of fields in the request/response header (RH) violates architectural rules or previously selected BIND options.

Category and description (in hexadecimal numbers):

4001 Session control (SC) or data flow control (DFC) request/response header (RH) not valid: The RH of an SC or DFC request was not valid. For example, an SC RH with pacing request indicator set to 1 is not valid.

4005 Incomplete RH: Transmission shorter than full transmission header-request/response header (TH-RH).

4007 Definite response not allowed: Definite response was requested when it was not permitted.

400B Chaining not supported: The begin and end chaining indicators (BCI and ECI) were not specified correctly. For example, chaining bits were indicated other than BCI or ECI, but multiple request chains are not supported for the session or for the category specified in the request header.

400C Brackets not supported: A bracket indicator, begin bracket indicator (BBI), conditional end bracket indicator (CEBI), or end bracket indicator (EBI) was specified. Brackets are not allowed.

400D Change direction (CD) not supported: The CD indicator was set but not supported.

400F Incorrect use of format indicator: The format indicator (FI) was not specified correctly. For example, FI was set with BCI = ¬BC, or FI was not set on a data flow control (DFC) request.

4010 Alternative code not supported: The code selection indicator (CSI) was set but is not supported.

4011 Incorrect specification of RU category: The RU category was not specified correctly. For example, an expedited-flow request or response was specified with the RU category indicator = FMD.

4014 Use of DR1I, DR2I, or ERI not correct: The definite response 1 indicator (DR1I), definite response 2 indicator (DR2I), and exception response indicator (ERI) were not specified correctly. For example, a SIGNAL request was not specified with DR1I = DR1, DR2I = ¬DR2, and ERI = ¬ER.

4015 Use of QRI not correct: The queued response indicator (QRI) was not specified correctly, for example, QRI = QR on an expedited-flow request.

4016 Use of EDI not correct: The enciphered data indicator (EDI) was not specified correctly, for example, EDI = ED on an expedited-flow request.

4017 Use of PDI not correct: The padded data indicator (PDI) was not specified correctly, for example, PDI = PD on an expedited.

Path Error (Category Code = X'80')

This category indicates that the request was not delivered to the intended receiver because of a break in the path, a sequence of activation requests that was not valid, or an error in one of the path information units. Some PIU errors are included into other categories; for example, sequence number errors are sense code category X'20'. A path error received while the session is active generally indicates that the path to the session partner was lost.

Category and description (in hexadecimal numbers):

8004 Destination not recognized: A node in the path has no routing information for the destination specified by the transmission header (TH).

8005 No session: A request that was not valid was received when no LU-LU session was active.

Bytes 2 and 3 following the sense code contain sense code specific information. Allowed settings are:

0000 No specific code applies

0001 The receiver received a request other than a session control request when no LU-LU session was active.

0003 The receiver received a session control request other than a BIND or an UNBIND request when no LU-LU session was active.

8006 FID not valid: Format identification (FID) not valid for the receiving node.

8007 Segmenting error: Segmenting is not supported.

800B Transmission header (TH) not complete: Transmission received was shorter than a TH.

Appendix C. Mapping Intersystem Communications Function Operations to Systems Network Architecture Command

This appendix shows the association between some of the Systems Network Architecture (SNA) commands and ICF finance communications application operations. In Figure C-1 on page C-2, 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.

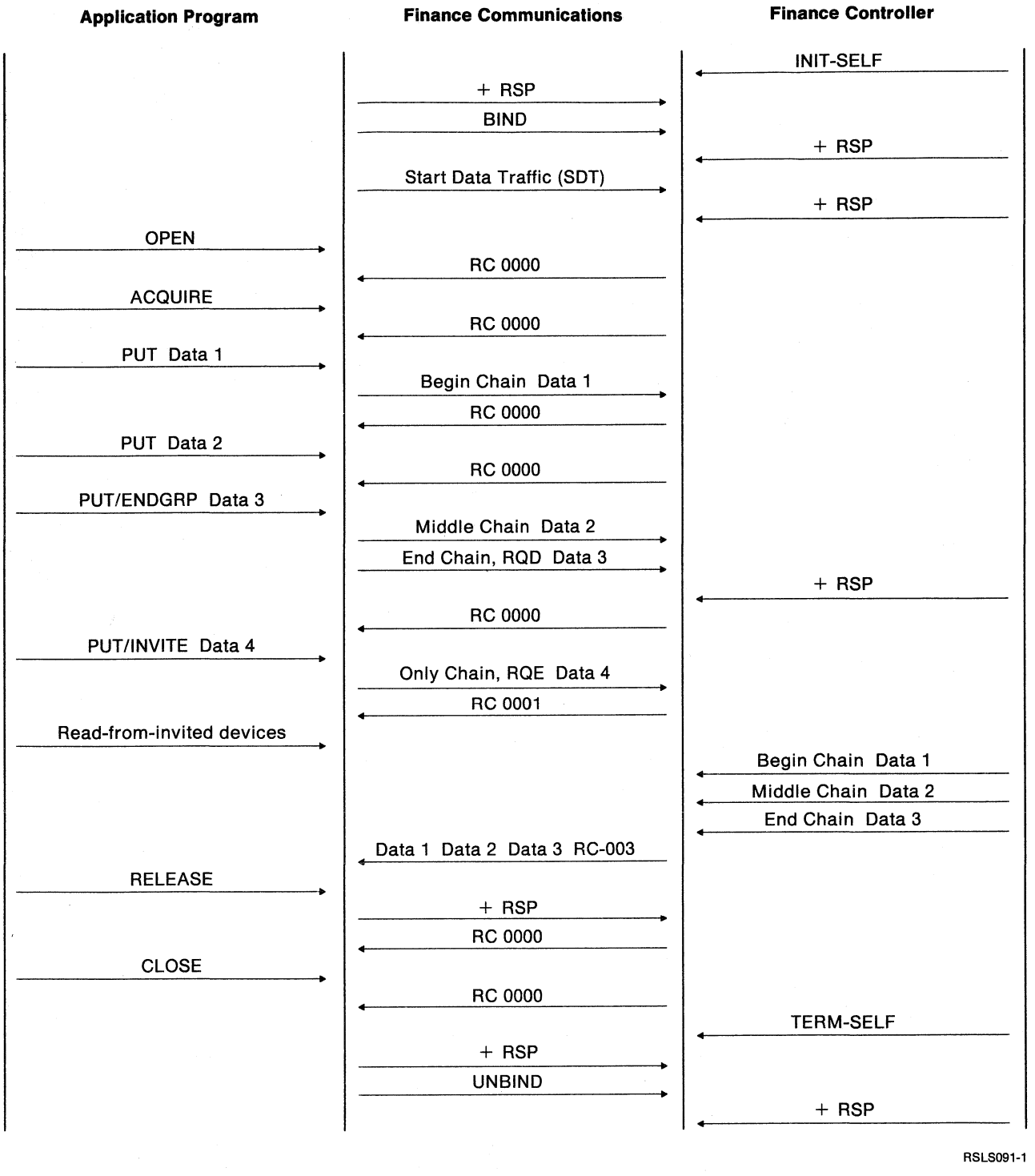


Figure C-1. SNA Commands with Corresponding ICF Application Operations

Appendix D. 4701 Finance Controller Diskette Download

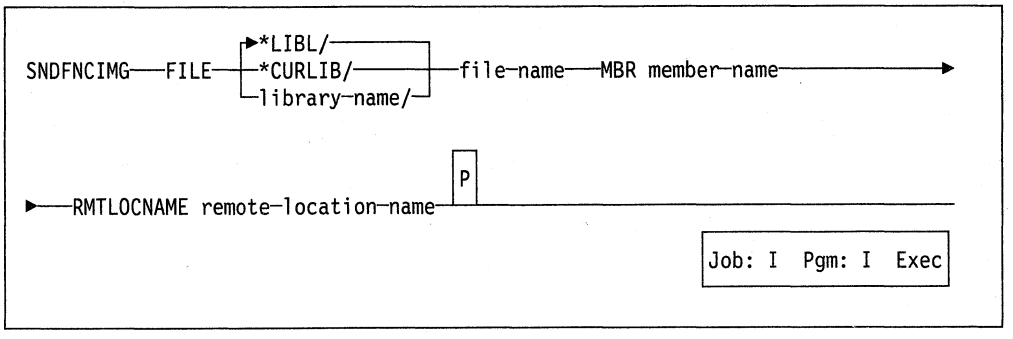
A 4701 finance controller must have an operating or initial program-load (IPL) diskette before it can be used. This diskette contains the basic license internal code, the controller configuration, and the application programs which can be created on a System/370 configured to support the 4700 finance communications system. The operating image created by the System/370 is loaded on the AS/400 system and the file will be sent to the 4701 controller using the system monitor session. This appendix describes the support which 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.

Member
Specifies the member in the file containing the diskette image which was blocked into a basic exchange format.

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.

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.

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.

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, 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. Figure D-1 shows the basic format of the diskette image and the basic exchange files.

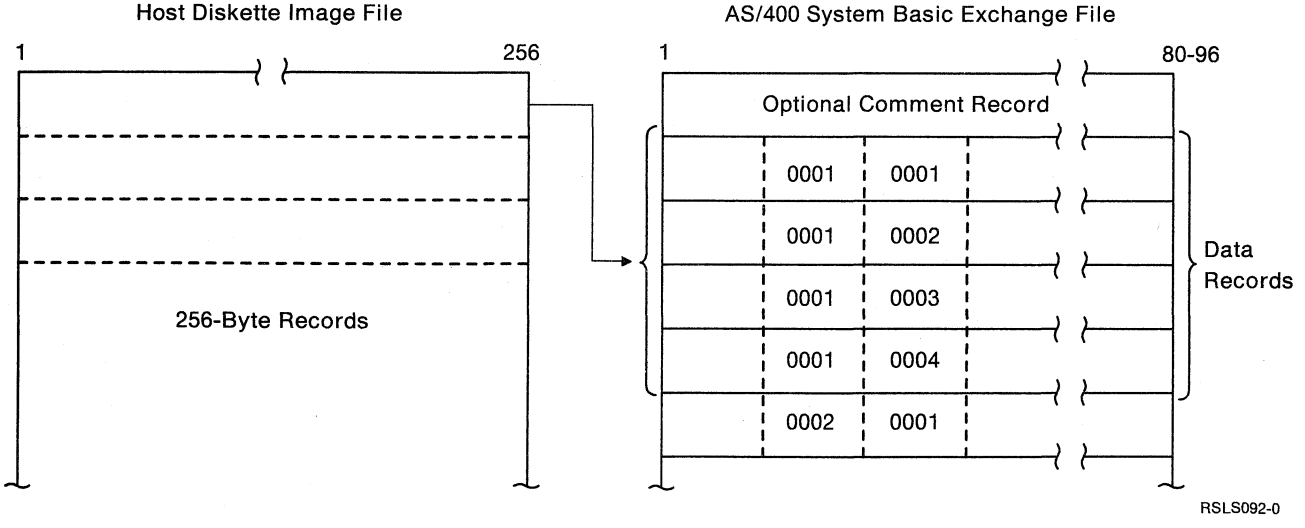


Figure D-1. Diskette Image Format and Basic Exchange Files

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 on page D-2.
3. Write the records to the basic exchange file. The required format for each data record is shown in Figure D-2.

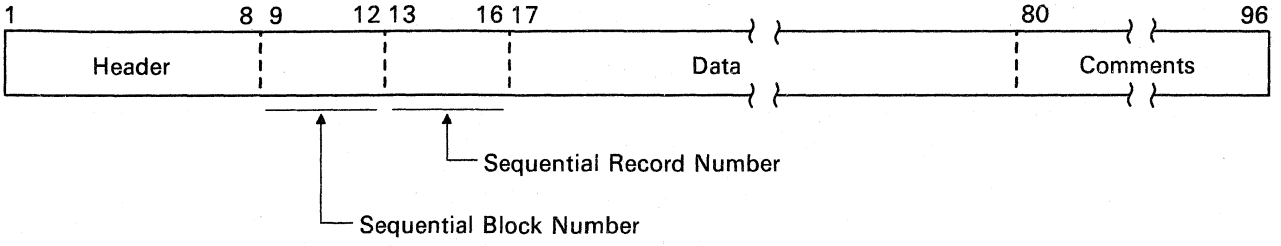


Figure D-2. Format Required for Data Records

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 *4700 Finance Communications System: Subsystem Operating Procedures*, and the *Host Support User's Guide*.

Appendix E. Intersystem Communications Function Finance Example Programs

This appendix provides COBOL/400 and RPG/400 example programs to demonstrate how finance communications is used. Both the following example programs are shown in Figure E-1.

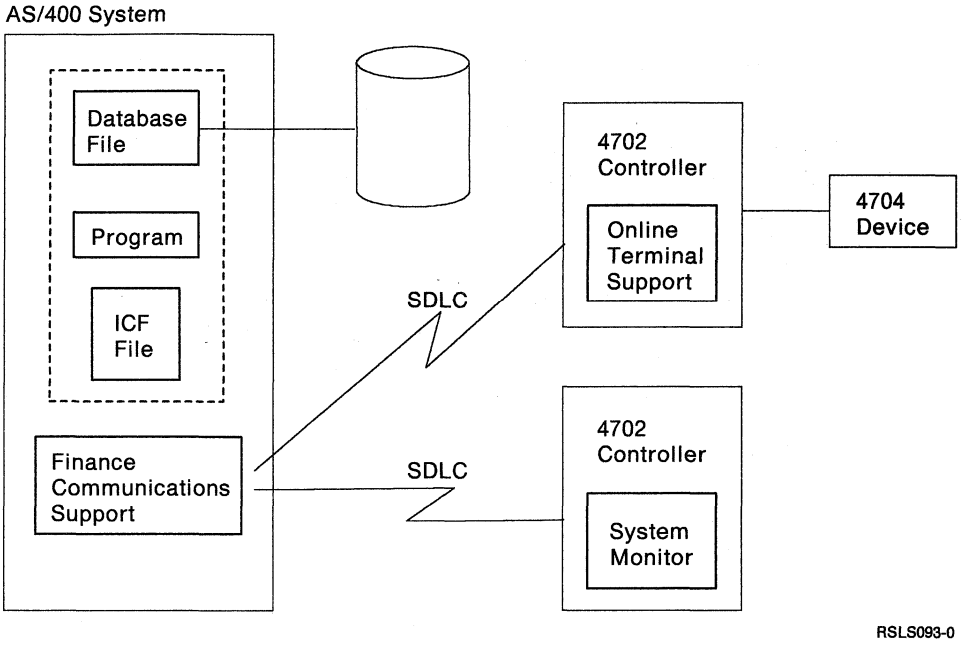


Figure E-1. Example for ICF Finance Network

COBOL/400 Source Program for Local System

The following subtopic describes the objects needed on the local system to run the 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) CMNTYPE(*FINANCE)
```

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)
```

```
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001ICF          06/05/89  9:13:01        Page    1
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
Authority . . . . . :   *CHANGE
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
                        * * * * *  E N D  O F  S O U R C E  * * * * *
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001ICF          06/05/89  9:13:01        Page    2
                                       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  * * * * *
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001ICF          06/05/89  9:13:01        Page    3
                                          Messages

   ID      Severity  Number
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001ICF          06/05/89  9:13:01        Page    4
                                       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. DDS Source for ICF File K001ICF

```
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001ICF         06/05/89  9:13:01        Page     1
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
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 E-3 (Part 1 of 2). DDS Source for Database File K001DBF

```
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001DBF          06/03/89  9:13:47       Page    2
                                       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  * * * * *

5728SS1 R02 M00 891006                Data Description              FNCLIB/K001DBF          06/03/89  9:13:47       Page    3
                                           Messages
   ID      Severity  Number
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001DBF          06/03/89  9:13:47       Page    4
                                       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

```
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001PRT         06/05/89  9:13:03        Page    1
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
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 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  * * * * *
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001PRT         06/05/89  9:13:03        Page    2
                                       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  0    21                                       10           1
   160                                             5'RECORD FORMAT:' SPACEB(2)             14
   170                   FMTNM          8A  0    20                                        8          11
   180                                             5'MAJOR CODE:' SPACEB(2)                11
   190                   MAJOR          2A  0    17                                        2          19
   200                                             5'MINOR CODE:' SPACEB(2)                11
   210                   MINOR          2A  0    17                                        2          21
              * * * * *  E N D  O F  E X P A N D E D  S O U R C E  * * * * *
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001PRT         06/05/89  9:13:03        Page    3
                                          Messages
   ID      Severity  Number
5728SS1 R02 M00 891006                Data Description              FNCLIB/K001PRT         06/05/89  9:13:03        Page    4
                                       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 topic explains the COBOL/400 account inquiry program example in Figure E-5 on page E-8.

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 On-line Terminal Support (OTS) application program on the 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 (by passing control to **11**) the error message to the finance controller.

9 This routine is called to build and send (by passing control to **11**) the customer information to the finance controller.

10 This routine is called to build and send (by passing control to **11**) the account information to the finance controller.

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 the topic "Responses" on page 7-2.

12 This routine is called to end the program. The session is ended by releasing the finance device and the files are closed.

```
5728CB1 R02 M00 891006              IBM AS/400  COBOL/400           FNCLIB/K001            06/05/89 16:10:11       Page    1
Program name . . . . . . . . . . . . : K001 in FNCLIB
Source file  . . . . . . . . . . . . : PGMSRC in FNCLIB     Member - K001     06/05/89 16:09:10
Compiler option  . . . . . . . . . . : *NONE
Code generation option . . . . . . . : *NONE
Code generation severity level . . . : 29
Print file . . . . . . . . . . . . . : QSYSPRT in *LIBL
FIPS flagging option . . . . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . : *NOFLAG
Flagging level . . . . . . . . . . . : 0
Replace existing program . . . . . . : *YES
Target release . . . . . . . . . . . : *CURRENT
User profile . . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . . : *BLANK
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
```

Figure E-5 (Part 1 of 12). COBOL/400 Program

```
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000010 IDENTIFICATION DIVISION.
       000020
    2  000030 PROGRAM-ID.  K001.
       000040
       000050*****************************************************************
       000060*                                                               *
       000070*     PROGRAM TO HANDLE ACCOUNT INQUIRY FROM A FINANCE          *
       000080*     CONTROLLER.                                               *
       000090*                                                               *
       000100*****************************************************************
       000110
    3  000120 ENVIRONMENT DIVISION.
       000130
    4  000140 CONFIGURATION SECTION.
    5  000150 SOURCE-COMPUTER.  IBM-AS400.
    6  000160 OBJECT-COMPUTER.  IBM-AS400.
       000170
    7  000180 SPECIAL-NAMES.    I-O-FEEDBACK IS FEEDBACK-AREA.
       000190
    8  000200 INPUT-OUTPUT SECTION.
1   9  000210 FILE-CONTROL.
       000220
   10  000230     SELECT K001DBF
   11  000240          ASSIGN TO DATABASE-K001DBF
   12  000250          ORGANIZATION IS INDEXED
   13  000260          ACCESS MODE IS RANDOM
   14  000270          RECORD KEY IS EXTERNALLY-DESCRIBED-KEY.
       000280
   15  000290     SELECT K001ICF
   16  000300          ASSIGN TO WORKSTATION-K001ICF-SI
   17  000310          ORGANIZATION IS TRANSACTION
   18  000320          CONTROL-AREA IS TRAN-CTL-AREA
   19  000330          FILE STATUS IS STATUS-IND MAJ-MIN.
       000340
   20  000350     SELECT K001PRT
   21  000360          ASSIGN TO FORMATFILE-K001PRT
   22  000370          ORGANIZATION IS SEQUENTIAL.
       000380
   23  000390 DATA DIVISION.
       000400
   24  000410 FILE SECTION.
       000420
   25  000430 FD  K001DBF
   26  000440     LABEL RECORDS ARE STANDARD.
   27  000450 01  ACCOUNT-REC.
   28  000460     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
```

Figure E-5 (Part 2 of 12). COBOL/400 Program

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/K001          06/05/89 16:10:11        Page     3
 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
      000470
   49  000480 FD  K001ICF
   50  000490     LABEL RECORDS ARE STANDARD.
   51  000500 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
      000510
   58  000520 FD  K001PRT
   59  000530     DATA RECORD IS PRT-REC.
   60  000540 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 LIBRARY FNCLIB          <-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
      000550
   67  000560 WORKING-STORAGE SECTION.
      000570
   68  000580 77  STATUS-IND      PIC XX.
   69  000590 77  INDON           PIC 1 VALUE B"1".
   70  000600 77  INDOFF          PIC 1 VALUE B"0".
      000610*****************************************************************
      000620*  PROGRAM DEVICES                                              *
      000630*****************************************************************
   71  000640 77  WS-PGMDEV       PIC X(10) VALUE "FNCTRGT   ".
      000650
   72  000660 01  PGM-INDIC-AREA.
   73  000670     05 PGM-INDIC    PIC 1 OCCURS 99 TIMES
```

Figure E-5 (Part 3 of 12). COBOL/400 Program

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING          FNCLIB/K001          06/05/89 16:10:11      Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
   74  000680                    INDICATOR 1.
       000690
   75  000700 01  TRAN-CTL-AREA.
   76  000710     05  FILLER        PIC X(2).
   77  000720     05  PGM-DEV-NAME  PIC X(10).
   78  000730     05  RCD-FMT-NAME  PIC X(10).
       000740
   79  000750 01  FEEDBACK-DATA.
   80  000760     05  FILLER        PIC X(37).
   81  000770     05  FMTNM         PIC X(10).
   82  000780     05  FILLER        PIC X(225).
   83  000790     05  PGMDEV        PIC X(10).
   84  000800     05  FILLER        PIC X(84).
   85  000810     05  FILLER        PIC X(34).
   86  000820     05  MAJOR         PIC X(2).
   87  000830     05  MINOR         PIC X(2).
       000840
   88  000850 01  MAJ-MIN.
   89  000860     05  MAJ           PIC X(2).
   90  000870     05  MIN           PIC X(2).
       000880
   91  000890 01  FILLER.
   92  000900     05    FFFFFF PIC 9(8) COMP-4 VALUE 16777215.
   93  000910     05    FILLER REDEFINES FFFFFF.
   94  000920          10  FILLER PIC X.
   95  000930          10  FFS    PIC XXX.
       000940
       000950*****************************************************************
       000960* FORMAT OF THE DATA SENT BY THE FINANCE CONTROLLER.            *
       000970*****************************************************************
   96  000980 01       FNC-INPUT-DATA.
   97  000990     10      FNC-CTL-ITEMS.
   98  001000          15        WSTYPE         PIC XX.
   99  001010          15        CTLUNIT        PIC XX.
  100  001020          15        WSNO           PIC XX.
  101  001030          15        AUDTNO         PIC XX.
  102  001040          15        TELLERNO       PIC 999.
  103  001050          15        LINENBR        PIC 99.
  104  001060          15        TRNCDE         PIC XXX.
  105  001070          15        SPLFNCT        PIC X.
  106  001080          15        RESVRD         PIC XXX.
  107  001090          15        CTLUTYPE       PIC X.
  108  001100     10     FNC-DATA-AREA.
  109  001110          15        FLD1P          PIC X.
  110  001120          15        FLD2P          PIC X.
  111  001130          15        FLD3P          PIC X.
  112  001140          15        FLD4P          PIC X.
  113  001150          15        FLD5P          PIC X.
  114  001160          15        FLD6P          PIC X.
  115  001170          15        FLD7P          PIC X.
  116  001180          15        INPUT-FIELD    PIC X(10) OCCURS 7 TIMES
  117  001190                                              INDEXED BY FLD.
       001200
       001210*****************************************************************
```

Figure E-5 (Part 4 of 12). COBOL/400 Program

```
      001220* FORMAT OF THE DATA SENT BY THE AS/400 TO THE FINANCE      *
5728CB1 R02 M00 891006               COBOL SOURCE LISTING            FNCLIB/K001            06/05/89 16:10:11       Page     5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      001230* CONTROLLER.                                              *
      001240*****************************************************************
  118 001250 01  FNC-OUTPUT-DATA.
  119 001260     05     FNC-CONTROL.
      001270*****************************************************************
      001280* THE TERMINAL-MODE VARIABLE PUTS THE 4704 DISPLAY INTO     *
      001290* LARGE SCREEN MODE WHEREBY 1920 CHARACTERS ARE DISPLAYED,  *
      001300* THEREBY ALLOWING MORE INFORMATION TO BE DISPLAYED, WHEN   *
      001310* THE VARIABLE IS SET TO 1.                                 *
      001320*****************************************************************
  120 001330          10        TERMINAL-MODE  PIC 9     VALUE IS 0.
  121 001340          10        FILLER         PIC X(4)  VALUE SPACES.
      001350*****************************************************************
      001360* THE MORE-DATA VARIABLE INFORMS THE FINANCE CONTROLLER THAT*
      001370* MORE DATA IS YET TO COME (THE TRANSACTION HAS NOT         *
      001380* COMPLETED), WHEN THE VARIABLE IS SET TO 1.                *
      001390*****************************************************************
  122 001400          10        MORE-DATA      PIC 9     VALUE IS 0.
  123 001410          10        FILLER         PIC X(10) VALUE SPACES.
  124 001420     05     FNC-FORMATTED-DATA     PIC X(240).
      001430
      001440*****************************************************************
      001450* DATA FORMAT FOR ERROR MESSAGE DISPLAY. THE FOLLOWING      *
      001460* ERRORS ARE HANDLED: ACCOUNT NUMBER AND ACCOUNT NOT FOUND  *
      001480*****************************************************************
  125 001490 01  FORMATTED-DATA-1.
  126 001500     05   NEW-PAGE-1     PIC X.
  127 001510     05   TEXT-1         PIC X(30).
  128 001520     05   ACCTNR         PIC ZZZZZZZZ.
  129 001530     05   DATA-SEP-1     PIC X.
  130 001540     05   LASTFF-1       PIC XXX.
      001550
      001560*****************************************************************
      001570* DATA FORMAT FOR FIRST HALF OF VALID INQUIRY DISPLAY.      *
      001580*****************************************************************
  131 001590 01  FORMATTED-DATA-2.
  132 001600     05   NEW-PAGE-2     PIC X.
  133 001610     05   SETPOS-2A      PIC X.
  134 001620     05   MOVHOR-2A      PIC X.
  135 001630     05   HEX20-2A       PIC X.
  136 001640     05   DATE-2         PIC ZZ/ZZ/ZZ.
  137 001650     05   NEWLIN-2B      PIC X.
  138 001660     05   FILLER         PIC X(5) VALUE "ACC#:".
  139 001670     05   ACCTNR         PIC ZZZZZZZ9.
  140 001680     05   FILLER         PIC X.
  141 001690     05   NAME           PIC X(21).
  142 001700     05   NEWLIN-2C      PIC X.
  143 001710     05   SETPOS-2C      PIC X.
  144 001720     05   MOVHOR-2C      PIC X.
  145 001730     05   HEX0F-2C       PIC X.
  146 001740     05   STR1           PIC XXX.
  147 001750     05   STR2           PIC X(18).
  148 001760     05   NEWLIN-2D      PIC X.
  149 001770     05   SETPOS-2D      PIC X.
```

Figure E-5 (Part 5 of 12). COBOL/400 Program

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING          FNCLIB/K001          06/05/89 16:10:11       Page    6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
  150  001780     05   MOVHOR-2D      PIC X.
  151  001790     05   HEX0F-2D       PIC X.
  152  001800     05   CITY           PIC X(16).
  153  001810     05   NEWLIN-2E      PIC X.
  154  001820     05   SETPOS-2E      PIC X.
  155  001830     05   MOVHOR-2E      PIC X.
  156  001840     05   HEX0F-2E       PIC X.
  157  001850     05   OCUP           PIC X(21).
  158  001860     05   DATA-SEP-2     PIC X.
  159  001870     05   LASTFF-2       PIC XXX.
       001880
       001890*****************************************************************
       001900* DATA FORMAT FOR SECOND HALF OF VALID INQUIRY DISPLAY.         *
       001910*****************************************************************
  160  001920 01  FORMATTED-DATA-3.
  161  001930     05   SETPOS-3       PIC X.
  162  001940     05   SETLIN-3       PIC X.
  163  001950     05   HEX06-3        PIC X.
  164  001960     05   FILLER         PIC X(5) VALUE "BAL :".
  165  001970     05   TBAL           PIC ZZ,ZZZ,ZZZ.99-.
  166  001980     05   FILLER         PIC X(5).
  167  001990     05   FILLER         PIC X(10) VALUE "TOT.RSRVD:".
  168  002000     05   NEWLIN-3A      PIC X.
  169  002010     05   FILLER         PIC X(5) VALUE "DEP :".
  170  002020     05   DLYDEP         PIC ZZ,ZZZ,ZZZ.99.
  171  002030     05   FILLER         PIC X(3).
  172  002040     05   TOTRSV-3       PIC ZZ,ZZZ,ZZZ.99-.
  173  002050     05   NEWLIN-3B      PIC X.
  174  002060     05   FILLER         PIC X(5) VALUE "WTH :".
  175  002070     05   DLYWTH         PIC ZZ,ZZZ,ZZZ.99.
  176  002080     05   NEWLIN-3C      PIC X.
  177  002090     05   FILLER         PIC X(5) VALUE "LMT1:".
  178  002100     05   LIMIT1         PIC ZZ,ZZZ,ZZZ.99.
  179  002110     05   NEWLIN-3D      PIC X.
  180  002120     05   FILLER         PIC X(5) VALUE "LMT2:".
  181  002130     05   LIMIT2         PIC ZZ,ZZZ,ZZZ.99.
  182  002140     05   NEWLIN-3E      PIC X.
  183  002150     05   FILLER         PIC X(5) VALUE "W/B :".
  184  002160     05   WRKBAL-3       PIC ZZ,ZZZ,ZZZ.99.
  185  002170     05   NEWLIN-3F      PIC X.
  186  002180     05   LCKDSC         PIC X(38).
  187  002190     05   DATA-SEP-3     PIC X.
  188  002200     05   LASTFF-3       PIC XXX.
       002210
       002220*****************************************************************
       002230* DATA AREA TO DEFINE THE HEX VALUES NEEDED IN THE PROGRAM      *
       002240* TO SEND CONTROL CHARACTERS TO THE FINANCE CONTROLLER.         *
       002250*****************************************************************
  189  002260 01  DUMMY1.
       002270*    05   HEX0C          PIC X.
       002280*    05   HEX15          PIC X.
  190  002290     06   BIN1           PIC 9999 COMP-4 VALUE IS 3093.
       002300*    05   HEX08          PIC X.
       002310*    05   HEX34          PIC X.
  191  002320     06   BIN2           PIC 9999 COMP-4 VALUE IS 2100.
```

Figure E-5 (Part 6 of 12). COBOL/400 Program

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/K001            06/05/89 16:10:11       Page    7
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      002330*    05   HEX04          PIC X.
      002340*    05   HEXFF          PIC X.
  192 002350     06   BIN3           PIC 9999 COMP-4 VALUE IS 1279.
      002360*    05   HEX02          PIC X.
      002370*    05   HEX12          PIC X.
  193 002380     06   BIN4           PIC 9999 COMP-4 VALUE IS 530.
      002390*    05   HEX0D          PIC X.
      002400*    05   HEX25          PIC X.
  194 002410     06   BIN5           PIC 9999 COMP-4 VALUE IS 3365.
      002420*    05   HEX0F          PIC X.
      002430*    05   HEX20          PIC X.
  195 002440     06   BIN6           PIC 9999 COMP-4 VALUE IS 3872.
      002450*    05   HEX0A          PIC X.
      002460*    05   HEX06          PIC X.
  196 002470     06   BIN7           PIC 9999 COMP-4 VALUE IS 2566.
      002480*    05   DUMMY00        PIC X.
      002490*    05   LINENBR-HEX    PIC X.
  197 002500     06   BIN-LINENBR    PIC 9999 COMP-4.
      002510
      002520*****************************************************************
      002530* REDEFINES AREA TO GIVE THE HEX CONTROL CODES MEANINGFUL       *
      002540* NAMES.                                                        *
      002550*****************************************************************
  198 002560 01  DUMMY2                              REDEFINES DUMMY1.
  199 002570     05   NEWPAG         PIC X.
  200 002580     05   NEWLIN         PIC X.
  201 002590     05   MOVHOR         PIC X.
  202 002600     05   SETPOS         PIC X.
  203 002610     05   SETLIN         PIC X.
  204 002620     05   DTASEP         PIC X.
  205 002630     05   HEX02          PIC X.
  206 002640     05   HEX12          PIC X.
  207 002650     05   HEX0D          PIC X.
  208 002660     05   HEX25          PIC X.
  209 002670     05   HEX0F          PIC X.
  210 002680     05   HEX20          PIC X.
  211 002690     05   HEX0A          PIC X.
  212 002700     05   HEX06          PIC X.
  213 002710     05   DUMMY00        PIC X.
  214 002720     05   LINENBR-HEX    PIC X.
      002730
      002740*****************************************************************
      002750* THESE ARE THE MESSAGES USED BY THE PROGRAM.                   *
      002760*****************************************************************
  215 002770 01  MESSAGES.
  216 002780   05   MSG1      PIC X(30) VALUE "ACCOUNT NR. NOT FOUND".
  217 002790   05   MSG2      PIC X(30) VALUE "ACCOUNT NR. NOT PASSED".
      002800
  218 002810 01  ACCOUNT-STATUS PIC 9.
  219 002820     88   VALID-ACCOUNT VALUE IS 1.
      002830
  220 002840 01  EDATE          PIC  999999.
  221 002850 01  ACCTNO         PIC  S9(8).
  222 002860 01  WRKBAL         PIC  9(8)V99  COMP-3.
  223 002870 01  TOTRSV         PIC  9(8)V99  COMP-3.
```

Figure E-5 (Part 7 of 12). COBOL/400 Program

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/K001          06/05/89 16:10:11      Page    8
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      002880
      002890*****************************************************************
      002900* THESE ARE THE PARAMETERS PASSED TO THE PROGRAM BY THE     *
      002910* OTS PROGRAM ON THE FINANCE CONTROLLER.                    *
      002920*****************************************************************
  224 002930 LINKAGE SECTION.
  225 002940 01  DATA-PARM      PIC X(256).
  226 002950 PROCEDURE DIVISION USING DATA-PARM.
      002960
      002970*****************************************************************
      002980*                                                           *
      002990*  THE FOLLOWING DECLARATIVES SECTION IS AN ERROR ROUTINE    *
      003000*  THAT IS RUN WHEN AN ERROR OCCURS ON THE READ OR WRITE     *
      003010*  OF THE ICF FILE "K001ICF". THE ROUTINE MOVES DATA FROM    *
      003020*  THE I-O FEEDBACK AREA TO THE "ERROR" FORMAT OF THE PRINT  *
      003030*  FILE "K001PRT". THE PROGRAM IS TERMINATED RELEASING       *
      003040*  THE FINANCE DEVICE, AND CLOSING ALL FILES.                *
      003050*                                                           *
      003060*****************************************************************
2     003070 DECLARATIVES.
      003080 ERROR-SECTION SECTION.
      003090     USE AFTER EXCEPTION PROCEDURE ON I-O.
      003100 ERROR-PARAGRAPH.
  227 003110     ACCEPT FEEDBACK-DATA FROM FEEDBACK-AREA.
  228 003120     MOVE CORRESPONDING FEEDBACK-DATA TO ERRREC-O.
  229 003130     WRITE PRT-REC FORMAT IS "ERRREC".
      003140
  230 003150     DROP WS-PGMDEV FROM K001ICF.
      003160
  231 003170     CLOSE K001ICF
      003180           K001DBF
      003190           K001PRT.
      003200
  232 003210     STOP RUN.
      003220 END DECLARATIVES.
      003230
      003240 MAIN-PROGRAM SECTION.
3     003250 INITIALIZE-PROGRAM.
  233 003260     OPEN I-O     K001ICF.
  234 003270     OPEN I-O     K001DBF.
  235 003280     OPEN OUTPUT  K001PRT.
      003290
  236 003300     ACQUIRE WS-PGMDEV  FOR K001ICF.
      003310
  237 003320     ACCEPT EDATE FROM DATE.
  238 003330     MOVE ZEROES TO PGM-INDIC-AREA.
      003340
      003350*****************************************************************
      003360*                                                           *
      003370* THE DATA COMES INTO THE PROGRAM FROM THE FINANCE          *
      003380* CONTROLLER WITH UP TO SEVEN FIELDS. THERE ARE SEVEN FLAGS *
      003390* THAT INDICATE WHICH FIELDS ARE PRESENT:                   *
      003400*    IF FIELD ONE IS PRESENT, FLD1P IS "1", OTHERWISE       *
      003410*    IT IS " ".                                             *
      003420*    IF FIELD TWO IS PRESENT, FLD2P IS "2", OTHERWISE       *
```

Figure E-5 (Part 8 of 12). COBOL/400 Program

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/K001            06/05/89 16:10:11       Page    9
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      003430*    IT IS " ".                                                  *
      003440*    AND SO ON . . .                                             *
      003450* THE DATA IS LOADED SEQUENTIALLY INTO THE AVAILABLE FIELDS *
      003460* SO IF THE OPERATOR ENTERS FIELDS 1, 3, 5, AND 7, THE DATA *
      003470* WILL BE STORED IN INPUT FIELDS 1, 2, 3, AND 4. IT IS      *
      003480* NECESSARY TO TEST FOR THE PRESENCE OF THE FIELDS AND MOVE *
      003490* EACH INPUT FIELD INTO THE CORRECT FIELD IN THE PROGRAM.    *
      003500* SINCE THIS PROGRAM ONLY DEALS WITH ACCOUNT INQUIRIES, IF   *
      003510* AN ACCOUNT NUMBER IS NOT PASSED, A MESSAGE WILL BE SENT.   *
      003520*                                                            *
      003530**************************************************************
      003540
  239 003550     MOVE DATA-PARM TO FNC-INPUT-DATA.
4 240 003560     SET FLD TO 1.
      003570
  241 003580     IF FLD1P = "1"
      003590     THEN
  242 003600       MOVE INPUT-FIELD(FLD) TO ACCTNO
  243 003610       PERFORM PROCESS-TRANSACTION
      003620     ELSE
  244 003630       MOVE MSG2 TO TEXT-1
  245 003640       PERFORM BUILD-FORMAT-1.
      003650
  246 003660     PERFORM CLEAN-UP.
      003670
      003680
      003690**************************************************************
      003700*                                                            *
      003710* BEGIN PROCESSING THE TRANSACTION -                         *
      003720* A READ FROM THE DATABASE FILE IS DONE USING THE ACCOUNT    *
      003730* NUMBER AS THE KEY.  IF THE READ WAS SUCCESSFUL, CUSTOMER   *
      003740* AND ACCOUNT INFORMATION IS SENT BACK TO THE CONTROLLER.    *
      003750* IF THE READ WAS UNSUCCESSFUL, A MESSAGE STATING THAT THE   *
      003760* ACCOUNT NUMBER WAS NOT FOUND WILL BE SENT TO THE           *
      003770* CONTROLLER. THE FILES ARE THEN CLOSED AND THE PROGRAM      *
      003780* IS TERMINATED.                                             *
      003790*                                                            *
      003800**************************************************************
      003810 PROCESS-TRANSACTION.
  247 003820     MOVE 1 TO ACCOUNT-STATUS.
5 248 003830     MOVE ACCTNO TO ACCTNR OF ACCOUNT-REC.
  249 003840     READ K001DBF INVALID KEY MOVE 0 TO ACCOUNT-STATUS.
      003850
6 251 003860     IF VALID-ACCOUNT
      003870     THEN
  252 003880       PERFORM VALID-INQUIRY
      003890     ELSE
  253 003900       MOVE MSG1 TO TEXT-1
  254 003910       PERFORM BUILD-FORMAT-1.
      003920
      003930
      003940**************************************************************
      003950*                                                            *
      003960* THE PROCEDURE TO DO A VALID INQUIRY SENDS OUT THE DISPLAY  *
      003970* IN TWO RECORDS. THE FIRST RECORD HAS THE BASIC CUSTUMER    *
```

Figure E-5 (Part 9 of 12). COBOL/400 Program

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/K001          06/05/89 16:10:11       Page   10
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      003980* INFORMATION ACCOUNT, NAME, AND ADDRESS.  THE SECOND HAS   *
      003990* THE BALANCE INFORMATION.                                  *
      004000*                                                          *
      004010************************************************************
 7    004020 VALID-INQUIRY.
  255 004030     PERFORM BUILD-FORMAT-2.
  256 004040     COMPUTE WRKBAL = TBAL   OF ACCOUNT-REC
      004050             - DLYWTH OF ACCOUNT-REC
      004060             + DLYDEP OF ACCOUNT-REC
      004070             + LIMIT1 OF ACCOUNT-REC
      004080             + LIMIT2 OF ACCOUNT-REC.
  257 004090     COMPUTE TOTRSV = RSVCS1 + RSVCS2 + RSVRM1 + RSVRM2.
  258 004100     PERFORM BUILD-FORMAT-3.
      004110
      004120************************************************************
      004130*                                                          *
      004140* SEND A MESSAGE TO THE CONTROLLER.                        *
      004150*                                                          *
      004160************************************************************
 8    004170 BUILD-FORMAT-1.
  259 004180     MOVE ACCTNO TO ACCTNR OF FORMATTED-DATA-1.
  260 004190     MOVE 0 TO TERMINAL-MODE.
  261 004200     MOVE NEWPAG TO NEW-PAGE-1.
  262 004210     MOVE DTASEP TO DATA-SEP-1.
  263 004220     MOVE FFS TO LASTFF-1
      004230
  264 004240     MOVE FORMATTED-DATA-1 TO FNC-FORMATTED-DATA.
      004250
  265 004260     MOVE 59 TO LENREC OF VARREC-O.
  266 004270     PERFORM SEND-RESPONSE.
      004280
      004290************************************************************
      004300*                                                          *
      004310* SEND THE CUSTOMER INFORMATION TO THE CONTROLLER.         *
      004320*                                                          *
      004330************************************************************
 9    004340 BUILD-FORMAT-2.
  267 004350     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-2.
  268 004360     MOVE EDATE TO DATE-2.
      004370
  269 004380     MOVE 1 TO TERMINAL-MODE.
  270 004390     MOVE NEWPAG TO NEW-PAGE-2.
  271 004400     MOVE NEWLIN TO NEWLIN-2B, NEWLIN-2C, NEWLIN-2D, NEWLIN-2E.
  272 004410     MOVE SETPOS TO SETPOS-2A, SETPOS-2C, SETPOS-2D, SETPOS-2E.
  273 004420     MOVE MOVHOR TO MOVHOR-2A, MOVHOR-2C, MOVHOR-2C, MOVHOR-2E.
  274 004430     MOVE HEX20 TO HEX20-2A.
  275 004440     MOVE HEX0F TO HEX0F-2C, HEX0F-2D, HEX0F-2E.
  276 004450     MOVE DTASEP TO DATA-SEP-2.
  277 004460     MOVE FFS TO LASTFF-2
      004470
  278 004480     MOVE 1 TO MORE-DATA.
      004490
  279 004500     MOVE FORMATTED-DATA-2 TO FNC-FORMATTED-DATA.
      004510
  280 004520     MOVE 138 TO LENREC OF VARREC-O.
```

Figure E-5 (Part 10 of 12). COBOL/400 Program

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/K001          06/05/89 16:10:11       Page   11
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
  281  004530     PERFORM SEND-RESPONSE.
       004540
       004550*****************************************************************
       004560*                                                               *
       004570* SEND THE ACCOUNT INFORMATION TO THE CONTROLLER.               *
       004580*                                                               *
       004590*****************************************************************
10     004600 BUILD-FORMAT-3.
  282  004610     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-3.
  283  004620     MOVE WRKBAL TO WRKBAL-3.
  284  004630     MOVE TOTRSV TO TOTRSV-3.
  285  004640     MOVE 1 TO TERMINAL-MODE.
  286  004650     MOVE NEWLIN TO NEWLIN-3A, NEWLIN-3B, NEWLIN-3C, NEWLIN-3D,
       004660             NEWLIN-3E, NEWLIN-3F.
  287  004670     MOVE SETPOS TO SETPOS-3.
  288  004680     MOVE SETLIN TO SETLIN-3.
  289  004690     MOVE HEX06 TO HEX06-3
  290  004700     MOVE DTASEP TO DATA-SEP-3.
  291  004710     MOVE FFS TO LASTFF-3
       004720
  292  004730     MOVE 0 TO MORE-DATA.
  293  004740     MOVE FORMATTED-DATA-3 TO FNC-FORMATTED-DATA.
       004750
  294  004760     MOVE 208 TO LENREC OF VARREC-O.
  295  004770     PERFORM SEND-RESPONSE.
       004780
       004790*****************************************************************
       004800*                                                               *
       004810* RESPOND TO THE ACCOUNT INQUIRY.                               *
       004820*                                                               *
       004830*****************************************************************
11     004840 SEND-RESPONSE.
  296  004850     MOVE FNC-OUTPUT-DATA TO DTAREC OF VARREC-O.
       004860
  297  004870     WRITE ICFREC
       004880       FORMAT IS "VARREC"
       004890       TERMINAL IS WS-PGMDEV.
       004900
       004910*****************************************************************
       004920*                                                               *
       004930* TERMINATE PROGRAM.                                            *
       004940*                                                               *
       004950*****************************************************************
12     004960 CLEAN-UP.
  298  004970     DROP WS-PGMDEV FROM K001ICF.
       004980
  299  004990     CLOSE K001ICF
       005000           K001DBF
       005010           K001PRT.
       005020
  300  005030     STOP RUN.
       005040
                        * * * * *  E N D  O F  S O U R C E  * * * * *
```

Figure E-5 (Part 11 of 12). COBOL/400 Program

```
5728CB1 R02 M00 891006              COBOL MESSAGES                  FNCLIB/K001          06/05/89 16:10:11       Page   12
 STMT
                                    MESSAGE SUMMARY
 TOTAL    INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL(40-99)
    0          0              0               0               0                0
                  * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
504 source records read
43 copy records read
3 copy members processed
0 sequence errors
0 was the highest severity message issued
 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 12 of 12). COBOL/400 Program

RPG/400 Source Program for Local System

The following subtopics describe the objects needed on the local system to run the RPG/400 CPGEN download 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(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) CMNTYPE(*FINANCE)
```

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)
```

```
5728SS1 R02 M00 891006               Data Description              FNCLIB/CPGICF          06/05/89  9:12:53       Page    1
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
Authority . . . . . . . . . . . . . . . . . . . . . . . . :   *CHANGE
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  * * * * *
```

Figure E-6 (Part 1 of 2). DDS Source for ICF File CPGICF

```
5728SS1 R02 M00 891006               Data Description              FNCLIB/CPGICF          06/05/89  9:12:53       Page    2
                                      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  * * * * *
5728SS1 R02 M00 891006               Data Description              FNCLIB/CPGICF          06/05/89  9:12:53       Page    3
                                         Messages
   ID      Severity  Number
5728SS1 R02 M00 891006               Data Description              FNCLIB/CPGICF          06/05/89  9:12:53       Page    4
                                      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  * * * * *
          CRTPRTF FILE(FNCLIB/CPGPRT) SRCFILE(FNCLIB/QDDSSRC)
                SRCMBR(CPGPRT)
```

Figure E-6 (Part 2 of 2). DDS Source for ICF File CPGICF

```
5728SS1 R02 M00 891006                Data Description              FNCLIB/CPGDBF          06/03/89  9:13:39       Page    1
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
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
   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  * * * * *
5728SS1 R02 M00 891006                Data Description              FNCLIB/CPGDBF          06/03/89  9:13:39       Page    2
                               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  * * * * *
5728SS1 R02 M00 891006                Data Description              FNCLIB/CPGDBF          06/03/89  9:13:39       Page    3
                                  Messages
   ID      Severity  Number
5728SS1 R02 M00 891006                Data Description              FNCLIB/CPGDBF          06/03/89  9:13:39       Page    4
                               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

```
5728SS1 R02 M00 891006                Data Description              FNCLIB/CPGPRT          06/05/89  9:12:55        Page    1
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
Authority . . . . . . . . . . . . . . . . . . . . :   *CHANGE
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 PROGRAM        *              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  * * * * *
```

Figure E-8 (Part 1 of 2). DDS Source for Printer File CPGPRT

```
5728SS1 R02 M00 891006                Data Description              FNCLIB/CPGPRT          06/05/89  9:12:55        Page    2
                                       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  0    21                                        10           1
  1600                                             5'RECORD FORMAT:' SPACEB(2)              14
  1700                   FMTNM          8A  0    20                                         8          11
  1800                                             5'MAJOR CODE:' SPACEB(2)                 11
  1900                   MAJOR          2A  0    17                                         2          19
  2000                                             5'MINOR CODE:' SPACEB(2)                 11
  2100                   MINOR          2A  0    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  * * * * *
5728SS1 R02 M00 891006                Data Description              FNCLIB/CPGPRT          06/05/89  9:12:55        Page    3
                                          Messages
   ID      Severity  Number
5728SS1 R02 M00 891006                Data Description              FNCLIB/CPGPRT          06/05/89  9:12:55        Page    4
                                       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 (Part 2 of 2). DDS Source for Printer File CPGPRT

Program Explanation

The following topic explains the RPG/400 CPGEN file download program example in Figure E-9 on page E-25.

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 receive 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 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 to hold 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.

5 FILL00 is the name of the structure used by the program to initialize 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** (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 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.

```
 5728RG1 R02M00  891006            IBM AS/400 RPG/400                  FNCLIB/CPGPGM          06/05/89  09:16:47       Page        1
 Compiler . . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . :    FNCLIB/CPGPGM
   Source file  . . . . . . . . . . :    FNCLIB/PGMSRC
   Source member  . . . . . . . . . :    CPGPGM
   Source listing options . . . . . :    *SOURCE      *XREF       *GEN        *NODUMP     *NOSECLVL
   Generation options . . . . . . . :    *NOLIST      *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
   SAA flagging . . . . . . . . . . :    *NOFLAG
   Generation severity level  . . . :    9
   Print file . . . . . . . . . . . :    *LIBL/QSYSPRT
   Replace program  . . . . . . . . :    *NO
   Target release . . . . . . . . . :    *CURRENT
   User profile . . . . . . . . . . :    *USER
   Authority  . . . . . . . . . . . :    *CHANGE
   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
```

Figure E-9 (Part 1 of 7). RPG/400 Program for ICF Finance

```
 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
     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
```

Figure E-9 (Part 2 of 7). RPG/400 Program for ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                FNCLIB/CPGPGM         06/05/89  09:16:47       Page    3
SEQUENCE                                                                     IND    DO    LAST        PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE      LINE   ID
4  ICNSTDS       DS
   I                                         1   2 INTCMD
   I                                         1   1 B01
   I                                         2   2 B03
   I                                         3   4 TRMCMD
   I                                         3   3 B05
   I                                         4   4 B88
5  IFILL00       DS
   I                                         1 256 NULLS
   I                                         1   1 BF0
   I                                         1   2 BF1
   I                                         1   4 BF2
   I                                         1   8 BF3
   I                                         1  16 BF4
   I                                         1  32 BF5
   I                                         1  64 BF6
   I                                         1 128 BF7
   I                                         2   2 BS0
   I                                         3   4 BS1
   I                                         5   8 BS2
   I                                         9  16 BS3
   I                                        17  32 BS4
   I                                        33  64 BS5
   I                                        65 128 BS6
   I                                       129 256 BS7
   C****************************************************************

   C* START PROGRAM                                                 *
   C*                                                               *
   C*   FILES ARE IMPLICITLY OPENED, AND THE SYSTEM MONITOR         *
   C*   PROGRAM DEVICE IS ACQUIRED.                                 *
   C*                                                               *
   C****************************************************************
6  C           'PGMDEV'  ACQ  CPGICF
   C                     MOVEL'PGMDEV'  PGMDEV 10
   C****************************************************************
   C*                                                               *
   C* SET THE CONSTANTS NEEDED IN THE PROGRAM, AND SEND THE         *
   C* INITIALIZATION SEQUENCE TO THE FINANCE CONTROLLER.            *
   C*                                                               *
   C****************************************************************
7  C                     EXSR SETCON
   C                     MOVE INTCMD    DATA2
   C                     WRITEINIT
   C****************************************************************
   C*                                                               *
   C* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATABASE FILE *
   C* 'CPGDBF'.  WHEN THE FILE REACHES END OF FILE, INDICATOR 99 *
   C* IS TURNED ON. CONDITION N99 IS USED ON THE WRITE TO AVOID  *
   C* SENDING A BLANK RECORD.                                       *
   C*                                                               *
   C****************************************************************
8  C                     MOVE NULLS     WRKREC
```

Figure E-9 (Part 3 of 7). RPG/400 Program for ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                 FNCLIB/CPGPGM       06/05/89  09:16:47      Page        4
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 LINES WRITES OUT THE TERMINATION CODE '0588'X*
    9200  C* 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                       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 4 of 7). RPG/400 Program for ICF Finance

```
SEQUENCE                                                                IND   DO   LAST      PAGE   PROGRAM
5728RG1 R02M00  891006          IBM AS/400 RPG/400                FNCLIB/CPGPGM      06/05/89  09:16:47     Page      5
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*   INFORMATION REGARDING THE DOWNLOAD OPERATION DUE TO         *
   16100  C*   A RESPONSE WHICH 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 5 of 7). RPG/400 Program for ICF Finance

```
 5728RG1 R02M00  891006             IBM AS/400 RPG/400                      FNCLIB/CPGPGM        06/05/89  09:16:47       Page       6
 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 DATABASE FILE
 5728RG1 R02M00  891006             IBM AS/400 RPG/400                      FNCLIB/CPGPGM        06/05/89  09:16:47       Page       8
                          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 6 of 7). RPG/400 Program for ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                FNCLIB/CPGPGM       06/05/89  09:16:47      Page       9
       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  * * * * *
 5728RG1 R02M00  891006            IBM AS/400 RPG/400                FNCLIB/CPGPGM       06/05/89  09:16:47      Page      10
                      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 referred to.
* 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  * * * * *
 5728RG1 R02M00  891006            IBM AS/400 RPG/400                FNCLIB/CPGPGM       06/05/89  09:16:47      Page      11
                        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 7 of 7). RPG/400 Program for ICF Finance

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. Both the following example programs are shown in Figure F-1.

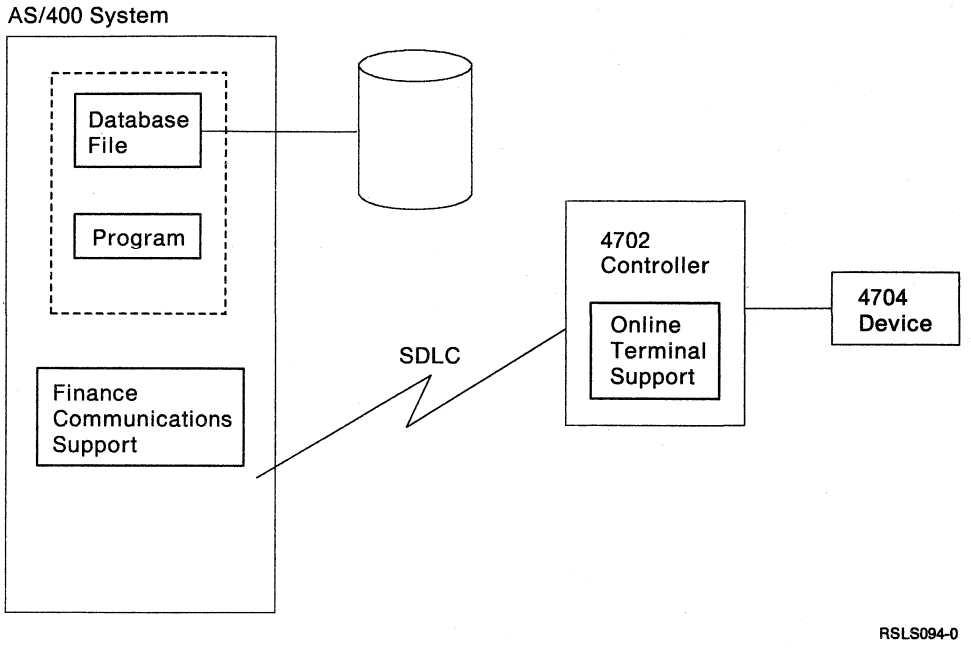


Figure F-1. Configuration Example for Non-ICF Finance

COBOL/400 and RPG/400 Source Programs for the Local System

The following subtopics describes the objects needed on the local system to run the COBOL/400 and 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)
           RMTLOCNAME(FNCDEV) 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)
```

The DDS used by this file is shown in Figure F-2 on page F-3.

```
          1. ACCOUNT
5728SS1 R02 M00 891006               Data Description              FNCLIB/ACCOUNT         06/03/89  9:13:47       Page     1
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

```
5728SS1 R02 M00 891006                Data Description              FNCLIB/ACCOUNT        06/03/89  9:13:47        Page     2
                                       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  * * * * *
5728SS1 R02 M00 891006                Data Description              FNCLIB/ACCOUNT        06/03/89  9:13:47        Page     3
                                          Messages
   ID      Severity  Number
5728SS1 R02 M00 891006                Data Description              FNCLIB/ACCOUNT        06/03/89  9:13:47        Page     4
                                       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

The following topic explains the COBOL/400 account inquiry and withdrawal program example in Figure F-3 on page F-7.

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 On-Line Terminal Support (OTS), 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 and valid request, and control is passed to either **6** (if the OTS the program is ended. Otherwise, the OTS request is a request is an account inquiry transaction) 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 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 length 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.

```
5728CB1 R02 M00 891006              IBM AS/400  COBOL/400          FNCLIB/OTSCBL1         06/07/89 09:41:18       Page     1
Program name . . . . . . . . . . . . : OTSCBL1 in FNCLIB
Source file  . . . . . . . . . . . . : PGMSRC in FNCLIB     Member - OTSCBL1     06/06/89 21:04:31
Compiler option  . . . . . . . . . . : *NONE
Code generation option . . . . . . . : *NONE
Code generation severity level . . . : 29
Print file . . . . . . . . . . . . . : QSYSPRT in *LIBL
FIPS flagging option . . . . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . : *NOFLAG
Flagging level . . . . . . . . . . . : 0
Replace existing program . . . . . . : *YES
Target release . . . . . . . . . . . : *CURRENT
User profile . . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . . : *BLANK
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
```

Figure F-3 (Part 1 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING           FNCLIB/OTSCBL1         06/07/89 09:41:18       Page     2
 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 FNC              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
   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
5728CB1 R02 M00 891006              COBOL SOURCE LISTING           FNCLIB/OTSCBL1         06/07/89 09:41:18       Page     3
```

Figure F-3 (Part 2 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
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 SYSTEM                     *
      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 SYSTEM 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 3 of 17). 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 4 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/OTSCBL1        06/07/89 09:41:18       Page    5
 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 5 of 17). 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 6 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING          FNCLIB/OTSCBL1        06/07/89 09:41:18       Page    7
 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 7 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/OTSCBL1        06/07/89 09:41:18       Page    8
 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 UP BY 1
      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
  244 003330       SET FLD UP BY 1
      003340     ELSE
  245 003350       MOVE ZEROS TO AMOUNT.
      003360
  246 003370     PERFORM PROCESS-TRANSACTION.
      003380
      003390************************************************************************
      003400* BEGIN PROCESSING THE TRANSACTION -                                   *
      003410* A READ OPERATION FROM THE DATABASE FILE IS DONE USING                *
      003420* THE ACCOUNT NUMBER AS THE KEY.  IF THE READ OPERATION WAS            *
      003430* SUCCESSFUL, PERFORM AN ACCOUNT INQUIRY OR A WITHDRAWAL               *
      003440* TRANSACTION (DEPENDING ON WHETHER A WITHDRAWAL AMOUNT WAS            *
      003450* PASSED).  OTHERWISE, SEND A MESSAGE TO THE FINANCE CONTROLLER        *
      003460* STATING THAT THE ACCOUNT NUMBER WAS NOT FOUND.                       *
      003470************************************************************************
      003480 PROCESS-TRANSACTION.
  247 003490     SET VALID-ACCOUNT TO TRUE.
  248 003500     MOVE ACCTNO TO ACCTNR OF ACCOUNT-REC.
[4] 249 003510     READ ACCOUNT INVALID KEY SET INVALID-ACCOUNT TO TRUE.
      003520
[5] 251 003530     IF VALID-ACCOUNT
      003540     THEN
  252 003550       IF AMOUNT > 0
      003560       THEN
  253 003570         PERFORM ATTEMPT-WITHDRAWAL
      003580       ELSE
  254 003590         PERFORM VALID-INQUIRY
      003600     ELSE
```

Figure F-3 (Part 8 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/OTSCBL1        06/07/89 09:41:18       Page     9
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
  255  003610       MOVE MSG(2) TO TEXT-1
  256  003620       PERFORM BUILD-FORMAT-1.
       003630
       003640*************************************************************************
       003650*                                                                       *
       003660* IF THE SNDLEN VALUE IS GREATER THAN 0, ONCE THE PROCESSING COMPLETES  *
       003670* AND THE PROGRAM ENDS, THE FINANCE JOB SENDS THE DATA IN THE           *
       003680* DATA-PARM BACK TO THE REQUESTING WORK STATION.  IF THE SNDLEN VALUE IS *
       003690* 0, THE FINANCE JOB SIMPLY SENDS AN INVITE TO THE WORK STATION TO      *
       003700* ALLOW THE OPERATOR TO INPUT THE NEXT TRANSACTION.                     *
       003710*                                                                       *
       003720*************************************************************************
       003730
  257  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* RECORD CONTAINS THE BALANCE INFORMATION.  BOTH SCREENS ARE SENT       *
       003870* USING THE IN-PROGRAM WRITE FUNCTION PROVIDED BY THE QFNWRT            *
       003880* ROUTINE.                                                              *
       003890*************************************************************************
       003900
6      003910 VALID-INQUIRY.
  258  003920     PERFORM BUILD-FORMAT-2.
  259  003930     CALL 'QFNWRT' USING WSID, SNDLEN, DATA-PARM, DATA-TYPE.
  260  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
  261  004000     COMPUTE TOTRSV = RSVCS1 + RSVCS2 + RSVRM1 + RSVRM2.
  262  004010     PERFORM BUILD-FORMAT-3.
       004020
  263  004030     CALL 'QFNWRT' USING WSID, SNDLEN, DATA-PARM, DATA-TYPE.
       004040
       004050*************************************************************************
       004060*                                                                       *
       004070* SET SNDLEN VALUE BACK TO 0 TO PREVENT THE FINANCE JOB FROM SENDING    *
       004080* ANY ADDITIONAL DATA FOR THIS TRANSACTION TO OTS ON THE                *
       004090* FINANCE CONTROLLER.
       004100*************************************************************************
       004110
  264  004120     COMPUTE SNDLEN = 0.
       004130
       004140*************************************************************************
       004150*                                                                       *
```

Figure F-3 (Part 9 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      004160* BEFORE PERFORMING A WITHDRAWAL, MAKE SURE THE ACCOUNT IS STILL       *
      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.
  265 004230     IF ACTIVE > 0
      004240     THEN
  266 004250       MOVE MSG(3) TO LCKDSC OF ACCOUNT-REC
  267 004260       PERFORM VALID-INQUIRY
      004270     ELSE
  268 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.
  269 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
  270 004460     COMPUTE TOTRSV = RSVCS1 + RSVCS2 + RSVRM1 + RSVRM2.
  271 004470     COMPUTE TOTAVL = WRKBAL - TOTRSV.
      004480
  272 004490     IF AMOUNT > TOTAVL
      004500     THEN
  273 004510       MOVE MSG(4) TO LCKDSC OF ACCOUNT-REC
  274 004520       PERFORM VALID-INQUIRY
      004530     ELSE
  275 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.
  276 004630     ADD AMOUNT TO WTHDRL OF ACCOUNT-REC,
      004640            DLYWTH OF ACCOUNT-REC.
  277 004650     SUBTRACT AMOUNT FROM TBAL OF ACCOUNT-REC.
  278 004660     PERFORM BUILD-FORMAT-4.
  279 004670     CALL 'QFNWRT' USING WSID, SNDLEN, DATA-PARM, DATA-TYPE.
  280 004680     REWRITE ACCOUNT-REC.
      004690
      004700************************************************************************
```

Figure F-3 (Part 10 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/OTSCBL1          06/07/89 09:41:18       Page   11
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      004710* SET SNDLEN BACK TO 0 SO THAT THE FINANCE JOB WILL NOT SEND ANY MORE   *
      004720* DATA FOR THIS TRANSACTION TO THE FINANCE FACILITY                     *
      004730*****************************************************************************
      004740
  281 004750     COMPUTE SNDLEN = 0.
      004760
      004770*****************************************************************************
      004780*                                                                           *
      004790* SEND AN ERROR MESSAGE 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.
  282 004850     MOVE ACCTNO TO ACCTNR OF FORMATTED-DATA-1.
  283 004860     MOVE 0 TO OTS-MODE.
  284 004870     MOVE NEWPAG TO NEW-PAGE-1.
  285 004880     MOVE DTASEP TO DATA-SEP-1.
  286 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
  287 004960     MOVE 0 TO MOREDT.
  288 004970     MOVE FORMATTED-DATA-1 TO OTS-FORMATTED-DATA.
  289 004980     MOVE OTS-OUTPUT-DATA TO DATA-PARM.
  290 004990     COMPUTE SNDLEN = 59.
      005000
 11   005010 BUILD-FORMAT-2.
  291 005020     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-2.
  292 005030     MOVE EDATE TO DATE-2.
  293 005040     MOVE 1 TO OTS-MODE.
  294 005050     MOVE NEWPAG TO NEW-PAGE-2.
  295 005060     MOVE NEWLIN TO NEWLIN-2B, NEWLIN-2C, NEWLIN-2D, NEWLIN-2E.
  296 005070     MOVE SETPOS TO SETPOS-2A, SETPOS-2C, SETPOS-2D, SETPOS-2E.
  297 005080     MOVE MOVHOR TO MOVHOR-2A, MOVHOR-2C, MOVHOR-2C, MOVHOR-2E.
  298 005090     MOVE HEX20 TO HEX20-2A.
  299 005100     MOVE HEX0F TO HEX0F-2C, HEX0F-2D, HEX0F-2E.
  300 005110     MOVE DTASEP TO DATA-SEP-2.
  301 005120     MOVE FFS TO LASTFF-2
      005130
      005140*****************************************************************************
      005150*  SET THE SEND LENGTH TO LENGTH OF THE FORMATTED DATA + 16.             *
      005160*****************************************************************************
      005170
  302 005180     MOVE 0 TO MOREDT.
  303 005190     MOVE FORMATTED-DATA-2 TO OTS-FORMATTED-DATA.
  304 005200     MOVE OTS-OUTPUT-DATA TO DATA-PARM.
  305 005210     COMPUTE SNDLEN = 138.
      005220
 12   005230 BUILD-FORMAT-3.
  306 005240     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-3.
  307 005250     MOVE WRKBAL TO WRKBAL-3.
```

Figure F-3 (Part 11 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/OTSCBL1         06/07/89 09:41:18       Page   12
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
  308  005260     MOVE TOTRSV TO TOTRSV-3.
  309  005270     MOVE 1 TO OTS-MODE.
  310  005280     MOVE NEWLIN TO NEWLIN-3A, NEWLIN-3B, NEWLIN-3C, NEWLIN-3D,
       005290          NEWLIN-3E, NEWLIN-3F.
  311  005300     MOVE SETPOS TO SETPOS-3.
  312  005310     MOVE SETLIN TO SETLIN-3.
  313  005320     MOVE HEX06 TO HEX06-3
  314  005330     MOVE DTASEP TO DATA-SEP-3.
  315  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
  316  005410     MOVE 0 TO MOREDT.
  317  005420     MOVE FORMATTED-DATA-3 TO OTS-FORMATTED-DATA.
  318  005430     MOVE OTS-OUTPUT-DATA TO DATA-PARM.
  319  005440     COMPUTE SNDLEN = 208.
       005450
 13    005460 BUILD-FORMAT-4.
  320  005470     MOVE CORRESPONDING ACCOUNTR TO FORMATTED-DATA-4.
  321  005480     MOVE EDATE TO DATE-4.
  322  005490     MOVE AMOUNT TO AMOUNT-4.
5728CB1 R02 M00 891006              COBOL SOURCE LISTING            FNCLIB/OTSCBL1         06/07/89 09:41:18       Page   13
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
  323  005500     MOVE WRKBAL TO WRKBAL-4.
  324  005510     MOVE TBAL OF ACCOUNT-REC TO BALOUT-4.
  325  005520     COMPUTE BIN-LINENBR = LINENBR + 1.
  326  005530     IF BIN-LINENBR > 18
  327  005540          COMPUTE BIN-LINENBR = 1.
  328  005550     MOVE 0 TO OTS-MODE.
  329  005560     MOVE NEWPAG TO NEWPAG-4.
  330  005570     MOVE NEWLIN TO NEWLIN-4B, NEWLIN-4C, NEWLIN-4D, NEWLIN-4E,
       005580          NEWLIN-4F, NEWLIN-4G, NEWLIN-4H, NEWLIN-4I.
  331  005590     MOVE SETPOS TO SETPOS-4A, SETPOS-4C.
  332  005600     MOVE MOVHOR TO MOVHOR-4A, MOVHOR-4C.
  333  005610     MOVE DTASEP TO DATA-SEP-4A.
  334  005620     MOVE HEX0F TO HEX0F-4C.
  335  005630     MOVE HEX20 TO HEX20-4A.
  336  005640     MOVE FFS TO LASTFF-4
  337  005650     MOVE 0 TO MOREDT.
  338  005660     MOVE FORMATTED-DATA-4 TO OTS-FORMATTED-DATA.
  339  005670     MOVE OTS-OUTPUT-DATA TO DATA-PARM.
  340  005680     COMPUTE SNDLEN = 190.
       005690
       005700*****************************************************************************
       005710*                                                                           *
       005720* TERMINATE PROGRAM.                                                        *
       005730*                                                                           *
       005740*****************************************************************************
 14    005750 CLEAN-UP.
       005760
  341  005770     CLOSE ACCOUNT.
       005780
  342  005790     STOP RUN.
       005800
                        * * * * *  E N D  O F  S O U R C E  * * * * *
```

Figure F-3 (Part 12 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006          COBOL DATA DIVISION MAP          FNCLIB/OTSCBL1          06/07/89 09:41:18        Page   14
 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   .D00556E
   21  02  ACCOUNTR                     FS  00000000      188  GROUP   .D0056B4
   22  03  ACCTNR                       FS  00000000        5  PACKED  .D005710
   23  03  NAME                         FS  00000005       21  AN      .D00577C
   24  03  STR1                         FS  00000026        3  AN      .D0057D4
   25  03  STR2                         FS  00000029       18  AN      .D00582C
   26  03  CITY                         FS  00000047       16  AN      .D005884
   27  03  ZIP                          FS  00000063        5  AN      .D0058DC
   28  03  OCUP                         FS  00000068       21  AN      .D005934
   29  03  TBAL                         FS  00000089        6  PACKED  .D00598C
   30  03  DLYWTH                       FS  00000095        6  PACKED  .D0059FA
   31  03  DLYDEP                      FS  00000101        6  PACKED  .D005A6A
   32  03  LIMIT1                       FS  00000107        6  PACKED  .D005ADA
   33  03  LIMIT2                       FS  00000113        6  PACKED  .D005B4A
   34  03  RSVCS1                       FS  00000119        6  PACKED  .D005BBA
   35  03  RSVCS2                       FS  00000125        6  PACKED  .D005C2A
   36  03  RSVRM1                       FS  00000131        6  PACKED  .D005C9A
   37  03  RSVRM2                       FS  00000137        6  PACKED  .D005D0A
   38  03  ACTIVE                       FS  00000143        1  PACKED  .D005D7A
   39  03  LCKDSC                       FS  00000144       38  AN      .D005DE6
   40  03  WTHDRL                       FS  00000182        6  PACKED  .D005E40
   42  01  FILLER                       WS  00000000        4  GROUP   .D005EB0
   43  02  FFFFFF                       WS  00000000        4  BINARY  .D005F0A  VALUE
   44  02  FILLER                       WS  00000000        4  GROUP   .D005F8E  REDEFINES .D005F0A
   45  03  FILLER                       WS  00000000        1  AN      .D005FE0
   46  03  FFS                          WS  00000001        3  AN      .D006032
   47  01  OTS-INPUT-DATA               WS  00000000       98  GROUP   .D0060C8
   48  02  OTS-CTL-ITEMS                WS  00000000       21  GROUP   .D00612A
   49  03  WSTYPE                       WS  00000000        2  AN      .D00618C
   50  03  CTLUNIT                      WS  00000002        2  AN      .D0061E6
   51  03  WSNO                         WS  00000004        2  AN      .D006242
   52  03  AUDTNO                       WS  00000006        2  AN      .D00629A
   53  03  TELLERNO                     WS  00000008        3  ZONED   .D0062F4
   54  03  LINENBR                      WS  00000011        2  ZONED   .D006360
   55  03  TRNCDE                       WS  00000013        3  AN      .D0063CA
   56  03  SPLFNCT                      WS  00000016        1  AN      .D006424
   57  03  RESVRD                       WS  00000017        3  AN      .D006480
   58  03  CTLUTYPE                     WS  00000020        1  AN      .D0064DA
   59  02  OTS-DATA-AREA                WS  00000021       77  GROUP   .D006536
   60  03  FLD1P                        WS  00000021        1  AN      .D006598
   61  03  FLD2P                        WS  00000022        1  AN      .D0065F2
   62  03  FLD3P                        WS  00000023        1  AN      .D00664C
   63  03  FLD4P                        WS  00000024        1  AN      .D0066A6
   64  03  FLD5P                        WS  00000025        1  AN      .D006700
   65  03  FLD6P                        WS  00000026        1  AN      .D00675A
   66  03  FLD7P                        WS  00000027        1  AN      .D0067B4
   67  03  INPUT-FIELD                  WS  00000028       10  AN      .D00680E  DIMENSION(7)
   68  IX  FLD                                                          .D006880
   69  01  OTS-OUTPUT-DATA              WS  00000000      256  GROUP   .D0068D8
   70  02  OTS-CONTROL                  WS  00000000       16  GROUP   .D00693C
   71  03  OTS-MODE                     WS  00000000        1  ZONED   .D00699C  VALUE
```

Figure F-3 (Part 13 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

STMT	LVL	SOURCE NAME	SECTION	DISP	LENGTH	TYPE	I-NAME	ATTRIBUTES
72	03	OTS-RSV	WS	00000001	4	AN	.D006A1A	VALUE
73	03	MOREDT	WS	00000005	1	ZONED	.D006A8C	VALUE
74	03	RSPDS	WS	00000006	1	AN	.D006B08	VALUE
75	03	DRSPF	WS	00000007	1	AN	.D006B76	VALUE
76	03	RSPMS	WS	00000008	1	AN	.D006BE4	VALUE
77	03	MRSPF	WS	00000009	1	AN	.D006C52	VALUE
78	03	RSPJP	WS	00000010	1	AN	.D006CC0	VALUE
79	03	JPRSPF	WS	00000011	1	AN	.D006D2E	VALUE
80	03	JPPARM	WS	00000012	1	AN	.D006D9C	VALUE
81	03	RSPPS	WS	00000013	1	AN	.D006E0A	VALUE
82	03	PPRSPF	WS	00000014	1	AN	.D006E78	VALUE
83	03	PSPARM	WS	00000015	1	AN	.D006EE6	VALUE
84	02	OTS-FORMATTED-DATA	WS	00000016	240	AN	.D006F54	
85	01	FORMATTED-DATA-1	WS	00000000	43	GROUP	.D006FBA	
86	02	NEW-PAGE-1	WS	00000000	1	AN	.D00701E	
87	02	TEXT-1	WS	00000001	30	AN	.D0070C8	
88	02	ACCTNR	WS	00000031	8	NE	.D007122	
89	02	DATA-SEP-1	WS	00000039	1	AN	.D007188	
90	02	LASTFF-1	WS	00000040	3	AN	.D0071E6	
91	01	FORMATTED-DATA-2	WS	00000000	122	GROUP	.D007242	
92	02	NEW-PAGE-2	WS	00000000	1	AN	.D0072A6	
93	02	SETPOS-2A	WS	00000001	1	AN	.D007304	
94	02	MOVHOR-2A	WS	00000002	1	AN	.D007362	
95	02	HEX20-2A	WS	00000003	1	AN	.D0073C0	
96	02	DATE-2	WS	00000004	8	NE	.D00741C	
97	02	NEWLIN-2B	WS	00000012	1	AN	.D00748A	
98	02	FILLER	WS	00000013	5	AN	.D0074E8	VALUE
99	02	ACCTNR	WS	00000018	8	NE	.D007552	
100	02	FILLER	WS	00000026	1	AN	.D0075B8	
101	02	NAME	WS	00000027	21	AN	.D00760A	
102	02	NEWLIN-2C	WS	00000048	1	AN	.D00765C	
103	02	SETPOS-2C	WS	00000049	1	AN	.D0076BA	
104	02	MOVHOR-2C	WS	00000050	1	AN	.D007718	
105	02	HEX0F-2C	WS	00000051	1	AN	.D007776	
106	02	STR1	WS	00000052	3	AN	.D0077D2	
107	02	STR2	WS	00000055	18	AN	.D007824	
108	02	NEWLIN-2D	WS	00000073	1	AN	.D007876	
109	02	SETPOS-2D	WS	00000074	1	AN	.D0078D4	
110	02	MOVHOR-2D	WS	00000075	1	AN	.D007932	
111	02	HEX0F-2D	WS	00000076	1	AN	.D007990	
112	02	CITY	WS	00000077	16	AN	.D0079EC	
113	02	NEWLIN-2E	WS	00000093	1	AN	.D007A3E	
114	02	SETPOS-2E	WS	00000094	1	AN	.D007A9C	
115	02	MOVHOR-2E	WS	00000095	1	AN	.D007AFA	
116	02	HEX0F-2E	WS	00000096	1	AN	.D007B58	
117	02	OCUP	WS	00000097	21	AN	.D007BB4	
118	02	DATA-SEP-2	WS	00000118	1	AN	.D007C06	
119	02	LASTFF-2	WS	00000119	3	AN	.D007C64	
120	01	FORMATTED-DATA-3	WS	00000000	192	GROUP	.D007CC0	
121	02	SETPOS-3	WS	00000000	1	AN	.D007D24	
122	02	SETLIN-3	WS	00000001	1	AN	.D007D80	
123	02	HEX06-3	WS	00000002	1	AN	.D007DDC	
124	02	FILLER	WS	00000003	5	AN	.D007E38	VALUE
125	02	TBAL	WS	00000008	14	NE	.D007EA2	
126	02	FILLER	WS	00000022	5	AN	.D007F0E	

Figure F-3 (Part 14 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006          COBOL DATA DIVISION MAP          FNCLIB/OTSCBL1          06/07/89 09:41:18          Page   16
 STMT LVL  SOURCE NAME              SECTION    DISP    LENGTH  TYPE    I-NAME    ATTRIBUTES
  127  02  FILLER                      WS  00000027        10  AN      .D007F60  VALUE
  128  02  NEWLIN-3A                   WS  00000037         1  AN      .D007FCE
  129  02  FILLER                      WS  00000038         5  AN      .D00802C  VALUE
  130  02  DLYDEP                      WS  00000043        13  NE      .D00080C8
  131  02  FILLER                      WS  00000056         3  AN      .D008134
  132  02  TOTRSV-3                    WS  00000059        14  NE      .D008186
  133  02  NEWLIN-3B                   WS  00000073         1  AN      .D0081FC
  134  02  FILLER                      WS  00000074         5  AN      .D00825A  VALUE
  135  02  DLYWTH                      WS  00000079        13  NE      .D0082C4
  136  02  NEWLIN-3C                   WS  00000092         1  AN      .D008330
  137  02  FILLER                      WS  00000093         5  AN      .D00838E  VALUE
  138  02  LIMIT1                      WS  00000098        13  NE      .D0083F8
  139  02  NEWLIN-3D                   WS  00000111         1  AN      .D008464
  140  02  FILLER                      WS  00000112         5  AN      .D0084C2  VALUE
  141  02  LIMIT2                      WS  00000117        13  NE      .D00852C
  142  02  NEWLIN-3E                   WS  00000130         1  AN      .D008598
  143  02  FILLER                      WS  00000131         5  AN      .D0085F6  VALUE
  144  02  WRKBAL-3                    WS  00000136        13  NE      .D008660
  145  02  NEWLIN-3F                   WS  00000149         1  AN      .D0086D6
  146  02  LCKDSC                      WS  00000150        38  AN      .D008734
  147  02  DATA-SEP-3                  WS  00000188         1  AN      .D008786
  148  02  LASTFF-3                    WS  00000189         3  AN      .D0087E4
  149  01  FORMATTED-DATA-4            WS  00000000       174  GROUP   .D008840
  150  02  NEWPAG-4                    WS  00000000         1  AN      .D0088A4
  151  02  SETPOS-4A                   WS  00000001         1  AN      .D008900
  152  02  MOVHOR-4A                   WS  00000002         1  AN      .D00895E
  153  02  HEX20-4A                    WS  00000003         1  AN      .D0089BC
  154  02  DATE-4                      WS  00000004         8  NE      .D008A18
  155  02  NEWLIN-4B                   WS  00000012         1  AN      .D008A86
  156  02  FILLER                      WS  00000013         5  AN      .D008AE4  VALUE
  157  02  ACCTNR                      WS  00000018         8  NE      .D008B4E
  158  02  FILLER                      WS  00000026         1  AN      .D008BB4
  159  02  NAME                        WS  00000027        21  AN      .D008C06
  160  02  NEWLIN-4C                   WS  00000048         1  AN      .D008C58
  161  02  SETPOS-4C                   WS  00000049         1  AN      .D008CB6
  162  02  MOVHOR-4C                   WS  00000050         1  AN      .D008D14
  163  02  HEX0F-4C                    WS  00000051         1  AN      .D008D72
  164  02  OCUP                        WS  00000052        21  AN      .D008DCE
  165  02  NEWLIN-4D                   WS  00000073         1  AN      .D008E20
  166  02  FILLER                      WS  00000074        10  AN      .D008E7E  VALUE
  167  02  AMOUNT-4                    WS  00000084        13  NE      .D008EEA
  168  02  NEWLIN-4E                   WS  00000097         1  AN      .D008F60
  169  02  NEWLIN-4F                   WS  00000098         1  AN      .D008FBE
  170  02  FILLER                      WS  00000099        10  AN      .D00901C  VALUE
  171  02  BALOUT-4                    WS  00000109        14  NE      .D0090C8
  172  02  NEWLIN-4G                   WS  00000123         1  AN      .D00913E
  173  02  FILLER                      WS  00000124        10  AN      .D00919C  VALUE
  174  02  WRKBAL-4                    WS  00000134        14  NE      .D00920A
  175  02  NEWLIN-4H                   WS  00000148         1  AN      .D009280
  176  02  NEWLIN-4I                   WS  00000149         1  AN      .D0092DE
  177  02  TEXT-4                      WS  00000150        20  AN      .D00933C
  178  02  DATA-SEP-4A                 WS  00000170         1  AN      .D009396
  179  02  LASTFF-4                    WS  00000171         3  AN      .D0093F6
  180  01  DUMMY1                      WS  00000000        16  GROUP   .D009452
  181  02  BIN1                        WS  00000000         2  BINARY  .D0094AC  VALUE
```

Figure F-3 (Part 15 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006              COBOL DATA DIVISION MAP          FNCLIB/OTSCBL1          06/07/89 09:41:18       Page   17
 STMT LVL  SOURCE NAME                 SECTION    DISP   LENGTH  TYPE   I-NAME    ATTRIBUTES
  182  02  BIN2                          WS  00000002        2  BINARY .D00952A  VALUE
  183  02  BIN3                          WS  00000004        2  BINARY .D0095A8  VALUE
  184  02  BIN4                          WS  00000006        2  BINARY .D009626  VALUE
  185  02  BIN5                          WS  00000008        2  BINARY .D0096A4  VALUE
  186  02  BIN6                          WS  00000010        2  BINARY .D009722  VALUE
  187  02  BIN7                          WS  00000012        2  BINARY .D0097A0  VALUE
  188  02  BIN-LINENBR                   WS  00000014        2  BINARY .D00981E
  189  01  DUMMY2                        WS  00000000       16  GROUP  .D00988E  REDEFINES .D009452
  190  02  NEWPAG                        WS  00000000        1  AN     .D0098E8
  191  02  NEWLIN                        WS  00000001        1  AN     .D009942
  192  02  MOVHOR                        WS  00000002        1  AN     .D00999C
  193  02  SETPOS                        WS  00000003        1  AN     .D0099F6
  194  02  SETLIN                        WS  00000004        1  AN     .D009A50
  195  02  DTASEP                        WS  00000005        1  AN     .D009AAA
  196  02  HEX02                         WS  00000006        1  AN     .D009B04
  197  02  HEX12                         WS  00000007        1  AN     .D009B5E
  198  02  HEX0D                         WS  00000008        1  AN     .D009BB8
  199  02  HEX25                         WS  00000009        1  AN     .D009C12
  200  02  HEX0F                         WS  00000010        1  AN     .D009C6C
  201  02  HEX20                         WS  00000011        1  AN     .D009CC6
  202  02  HEX0A                         WS  00000012        1  AN     .D009D20
  203  02  HEX06                         WS  00000013        1  AN     .D009D7A
  204  02  DUMMY00                       WS  00000014        1  AN     .D009DD4
  205  02  LINENBR-HEX                   WS  00000015        1  AN     .D009E30
  206  01  CONTROL-BYTE                  WS  00000000        1  GROUP  .D009E90
  207  02  DATA-TYPE                     WS  00000000        1  AN     .D009EF0  VALUE
  208  01  MESSAGES                      WS  00000000      120  GROUP  .D009F62
  209  02  MSG1                          WS  00000000       30  AN     .D009FBE  VALUE
  210  02  MSG2                          WS  00000030       30  AN     .D00A03E  VALUE
  211  02  MSG3                          WS  00000060       30  AN     .D00A0F0  VALUE
  212  02  MSG4                          WS  00000090       30  AN     .D00A172  VALUE
  213  01  MSG-TABLE                     WS  00000000      120  GROUP  .D00A1F8  REDEFINES .D009F62
  214  02  MSG                           WS  00000000       30  AN     .D00A256  DIMENSION(4)
  215  01  ACCOUNT-STATUS                WS  00000000        1  AN     .D00A2C0
  216  88  VALID-ACCOUNT                 WS
  217  88  INVALID-ACCOUNT               WS
  218  01  EDATE                         WS  00000000        6  ZONED  .D00A410
  219  01  ACCTNO                        WS  00000000        8  ZONED  .D00A47C
  220  01  AMOUNT                        WS  00000000        7  PACKED .D00A4E8
  221  01  AMOUNT-IN                     WS  00000000       10  ZONED  .D00A556
  222  01  WRKBAL                        WS  00000000        6  PACKED .D00A5C4
  223  01  TOTRSV                        WS  00000000        6  PACKED .D00A632
  224  01  TOTAVL                        WS  00000000        6  PACKED .D00A6A0
  226  01  WSID                          LS  00000001       10  AN     .D00A70E
  227  01  SNDLEN                        LS  00000002        8  PACKED .D00A766
  228  01  DATA-PARM                     LS  00000003      256  AN     .D00A7D6
   11      DB-FORMAT-NAME                SR  00001076       10  AN     .D00A894
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  * * * * *
```

Figure F-3 (Part 16 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

```
5728CB1 R02 M00 891006                    COBOL MESSAGES                FNCLIB/OTSCBL1          06/07/89 09:41:18       Page   18
 STMT
*  280  MSGID: LBL0412  SEVERITY: 20  SEQNBR:  004680
        Message . . . . :   INVALID KEY phrase not found in REWRITE
          statement. Accepted.
                                          MESSAGE SUMMARY
  TOTAL    INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL(40-99)
     1          0              0              1               0                0
                    * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
580 source records read
25 copy records read
1 copy members processed
0 sequence errors
20 was the highest severity message issued
 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 17 of 17). COBOL/400 Program OTSCBL1 for Non-ICF Finance

RPG/400 Program Explanation

The following topic explains the RPG/400 account inquiry and withdrawal program example in Figure F-4 on page F-23.

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 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 parameter.

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 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.

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                    FNCLIB/OTSRPG1        06/07/89  09:43:44       Page     1
Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . . . . . . . . :   FNCLIB/OTSRPG1
  Source file  . . . . . . . . . . :   FNCLIB/PGMSRC
  Source member  . . . . . . . . . :   OTSRPG1
  Source listing options . . . . . :   *SOURCE     *XREF       *GEN        *NODUMP     *NOSECLVL
  Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
  SAA flagging . . . . . . . . . . :   *NOFLAG
  Generation severity level  . . . :   9
  Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
  Replace program  . . . . . . . . :   *YES
  Target release . . . . . . . . . :   *CURRENT
  User profile . . . . . . . . . . :   *USER
  Authority  . . . . . . . . . . . :   *CHANGE
  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
```

Figure F-4 (Part 1 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                    FNCLIB/OTSRPG1        06/07/89  09:43:44       Page     2
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 CONTROLLER.                                                                                *
    1500  F*                                                                                                     *
    1600  E*********************************************************************                                 *
    1700  E**
    1800  E** TABLES
    1900  E**
           RECORD FORMAT(S):  LIBRARY FNC 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
 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
```

Figure F-4 (Part 2 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                      FNCLIB/OTSRPG1      06/07/89  09:43:44       Page      3
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 3 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      4
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 4 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006              IBM AS/400 RPG/400                     FNCLIB/OTSRPG1        06/07/89  09:43:44       Page      5
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 5 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                 FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      6
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 PROGRAM
   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 6 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                  FNCLIB/OTSRPG1       06/07/89  09:43:44      Page       7
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 7 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                 FNCLIB/OTSRPG1       06/07/89  09:43:44      Page       8
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 8 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                  FNCLIB/OTSRPG1      06/07/89  09:43:44      Page       9
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  * * * * *
```

Figure F-4 (Part 9 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
        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
5728RG1 R02M00  891006            IBM AS/400 RPG/400                  FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      10
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
5728RG1 R02M00  891006            IBM AS/400 RPG/400                  FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      11
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 10 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006           IBM AS/400 RPG/400                FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      12
                  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
5728RG1 R02M00  891006           IBM AS/400 RPG/400                FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      13
                    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)
       .PL001       PLIST     24000M   24000D
* 7031  *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
```

Figure F-4 (Part 11 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
                          23800M  24300   24800M  24900M  26100M
                          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
```

Figure F-4 (Part 12 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      15
       TOTRSV      P(10,2)   18400D   18500M   18600M   18700M   19900
                             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
* 7031 99           9800M   22600M   23600M   24700M   26000M
```

Figure F-4 (Part 13 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006            IBM AS/400 RPG/400                FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      16
     * * * * *  E N D   O F   C R O S S   R E F E R E N C E   * * * * *
5728RG1 R02M00  891006            IBM AS/400 RPG/400                FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      17
                        M e s s a g e   S u m m a r y
* QRG7031 Severity:  00   Number:   20
         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   * * * * *
5728RG1 R02M00  891006            IBM AS/400 RPG/400                FNCLIB/OTSRPG1      06/07/89  09:43:44      Page      18
                          F i n a l   S u m m a r y
 Message Count:  (by Severity Number)
           TOTAL    00     10     20     30     40     50
            20      20      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 14 of 14). Source for RPG/400 Program OTSRPG1 for Non-ICF Finance

```
5728RG1 R02M00  891006              IBM AS/400 RPG/400                      FNCLIB/SUBEDT           06/07/89  09:41:09       Page      1
Compiler . . . . . . . . . . . . . :    IBM AS/400 RPG/400
Command Options:
  Program  . . . . . . . . . . . . :    FNCLIB/SUBEDT
  Source file  . . . . . . . . . . :    FNCLIB/PGMSRC
  Source member  . . . . . . . . . :    SUBEDT
  Source listing options . . . . . :    *SOURCE      *XREF        *GEN         *NODUMP      *NOSECLVL
  Generation options . . . . . . . :    *NOLIST      *NOXREF      *NOATR       *NODUMP      *NOOPTIMIZE
  SAA flagging . . . . . . . . . . :    *NOFLAG
  Generation severity level  . . . :    9
  Print file . . . . . . . . . . . :    *LIBL/QSYSPRT
  Replace program  . . . . . . . . :    *NO
  Target release . . . . . . . . . :    *CURRENT
  User profile . . . . . . . . . . :    *USER
  Authority  . . . . . . . . . . . :    *CHANGE
  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
```

Figure F-5 (Part 1 of 2). RPG/400 Program SUBEDT for Non-ICF Finance

```
5728RG1 R02M00  891006              IBM AS/400 RPG/400                      FNCLIB/SUBEDT           06/07/89  09:41:09       Page      2
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 OUTOUT 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
5728RG1 R02M00  891006              IBM AS/400 RPG/400                      FNCLIB/SUBEDT           06/07/89  09:41:09       Page      3
                         C r o s s   R e f e r e n c e
Field References:
       FIELD        ATTR     REFERENCES (M=MODIFIED D=DEFINED)
* 7031  *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   * * * * *
5728RG1 R02M00  891006              IBM AS/400 RPG/400                      FNCLIB/SUBEDT           06/07/89  09:41:09       Page      4
                         M e s s a g e   S u m m a r y
* QRG7031 Severity:  00    Number:    1
        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   * * * * *
5728RG1 R02M00  891006              IBM AS/400 RPG/400                      FNCLIB/SUBEDT           06/07/89  09:41:09       Page      5
                         F i n a l   S u m m a r y
Message Count:  (by Severity Number)
           TOTAL     00      10      20      30      40      50
               1      1       0       0       0       0       0
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 (Part 2 of 2). RPG/400 Program SUBEDT for Non-ICF Finance

Glossary

access. To read; the ability to use or read.

acquire. To assign a display station or session to a program.

acquire-program-device operation. An operation that makes a program device available for input or output operations. Contrast with *release-program-device operation*.

active file. A tape or diskette file with an expiration date greater than the system date.

ACTLU. An SNA command used to start a session on a logical unit.

ACTPU. An SNA command used to start a session on a physical unit.

address. In data communications, the unique code assigned to the location of each device or system connected in a network.

advanced program-to-program communications (APPC). Data communications support that allows programs on an AS/400 system to communicate with programs on other systems having compatible communications support. APPC is the AS/400 method of using the SNA LU session type 6.2 protocol.

all authority. An object authority that allows the user to perform all operations on the object except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, and change the object. Contrast with *exclude authority*.

allocate. To reserve a resource for use in performing a specific task. Contrast with *deallocate*.

alphameric. Pertaining to the letters, A through Z or a through z; numbers, 0-9; and special symbols, $, #, @, ., or _. Synonymous with *alphanumeric*.

alphanumeric. Pertaining to the letters, A through Z or a through z; numbers, 0-9; and special symbols, $, #, @, ., or _. Synonymous with *alphameric*.

APPC. See *advanced program-to-program communications (APPC)*.

application. (1) A particular business task, such as inventory control or accounts receivable. (2) A program or set of programs that perform a task; for example, a payroll application.

application program. A program used to perform a particular data processing task such as inventory control or payroll.

AS/400 Cryptographic Support. The IBM licensed program that provides support for the encryption and decryption of data, according to the Data Encryption Algorithm, for managing cryptographic keys.

assumed value. A value supplied by the system when no value is specified by the user.

attribute. A characteristic or property of one or more objects.

automatic answer. In data communications, a line type that does not require operator action to receive a call over a switched line. Contrast with *manual answer*.

automatic call. A feature that permits a station to connect with another station over a switched line without operator action. Contrast with *manual call*.

automatic call unit. A common carrier device that allows the AS/400 system to automatically dial a remote location.

automatic dial. A function of the system that allows a system to automatically dial a remote station over a switched line without operator action.

automatic vary on. An option specified during the creation of configuration objects that allows them to be available when the system is started (IPL).

batch. Pertaining to a group of jobs to be run on a computer sequentially with the same program with little or no operator action. Contrast with *interactive*.

batch job. A predefined group of processing actions submitted to the system to be performed with little or no interaction between the user and the system. Contrast with *interactive job*.

binary synchronous communications (BSC). A data communications line protocol that uses a standard set of transmission control characters and control character sequences to send binary-coded data over a communications line. See also *synchronous data link control (SDLC)*.

bind command. A command used to start a session and define the characteristics of that session. Contrast with *unbind command*.

bracket. One or more chains of request units and their responses, representing a complete transaction,

exchanged between two logical unit half-sessions. See also *RU chain.*

BSC. See *binary synchronous communications (BSC).*

buffer. (1) A routine or an area of storage that corrects for the different speeds of data flow or timings of events, when transferring data from one device to another. (2) A portion of storage used to hold input or output data temporarily.

C language. A language used to develop application programs in compact, efficient code that can be run on different types of computers with minimal change.

C/400. The IBM licensed program that is the SAA C programming language available on the AS/400 system, including system-specific functions.

call level. The position of a program in a nest of programs called explicitly by the CALL instruction or implicitly by some event. The first program has a call level of 1. Any program called by a level 1 program has a call level of 2, and so on.

CCITT. The International Telegraph and Telephone Consultative Committee.

chain. (1) A group of logically linked records. (2) (BASIC) An operation in which a program passes control to another program, then ends. (3) (RPG/400) An operation code that reads input records identified by specified relative record numbers or keys. (4) (SNA) A group of logically linked records that are transferred over a communications line. See also *RU chain.*

chaining. 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.

character. Any letter, number, or other symbol in the data character set that is part of the organization, control, or representation of data.

character key. A keyboard key that allows the user to type into the system the character shown on the key. See also *function key*.

CL. See *control language (CL).*

class-of-service description. A system object created for advanced peer-to-peer networking (APPN) that provides the information required to assign relative priority to the transmission groups and intermediate routing nodes for an APPN session.

COBOL (common business-oriented language). A high-level programming language, based on English, that is used primarily for commercial data processing.

COBOL character. Any of the 51 characters of the COBOL character set.

COBOL/400. A licensed program that is a high-level programming language, resembling English. COBOL/400 is especially efficient in the processing of business problems.

column function. A process that calculates a value from a set of values and expresses it as a function name followed by an argument enclosed in parentheses.

command. (1) A statement used to request a function of the system. A command consists of the command name, which identifies the requested function and parameters. (2) (SNA) Any field set in the transmission header (TH), request header (RH), or a request unit that states an action or that starts a protocol.

command file. A remote job input stream that can contain host system commands and job control language (JCL), data, and RJE control statements (READFILE or EOF). Contrast with *data file.*

common user identification (common user ID). The user identification of a PC Support user that is used for the router entry in the CONFIG.PCS file or in the alternative configuration file if either file does not have a user ID specified. The common user ID of a PC Support user is the same on each host system that the router is connecting to the personal computer. See also *user identification (user ID).*

communications adapter. A part that electrically or physically connects a computer or device to a data communications network.

communications controller. The I/O processor card in the card enclosure.

communications line. The physical link (such as a wire or a telephone circuit) that connects one or more work stations to a communications controller unit, or connects one controller to another. Contrast with *data link protocol.*

communications manager. A function of the OS/2 Extended Edition program that lets a work station connect to a host computer and use the host resources as well as the resources of other personal computers to which the work station is attached, either directly or through a host. Communications manager provides application programming interfaces (APIs) so that users can develop their own applications.

communications security. A system option that requires the identity of a remote location to be verified before that location can run programs on your system.

communications type. A method for application programs to communicate on a local AS/400 system, or between a local AS/400 system and a remote system using the intersystem communications function (ICF).

Examples of these communications methods include (a) Systems Network Architecture (SNA) such as advanced program-to-program communications (APPC) and SNA upline facility (SNUF), (b) binary synchronous communications (BSC), and (c) asynchronous communications.

compilation. Translation of a source program (such as RPG/400 or COBOL specifications) into a program in machine language.

compile. To translate a program written in a high-level programming language into a machine-language program.

compiled program. The set of machine language instructions that is the output from the compilation of a source program. The actual processing of data is done by the machine-language program.

compiler. A program that translates programming language into machine language for use by the computer.

concurrent. Pertaining to the shared use of resources by multiple interactive users or application programs at the same time.

configuration. The physical and logical arrangement of devices and programs that make up a data processing system. See also *communications configuration*, *line configuration*, *controller configuration*, and *device configuration*.

configuration list. A list of local and remote locations and network addresses.

configure. To describe the interconnected arrangement of the devices, programs, communications, and optional features installed on a system.

consecutive processing. A method of processing in which the records in the file are read, written to, or deleted in the order in which they exist in a file. See also *random processing* and *sequential processing*.

consumer transaction facility (CTF). A stand-alone finance device used to handle transactions for banking customers. Synonymous with automatic teller machine (ATM).

contention state. In data communications, a type of half-duplex line or data link control in which either user may transmit any time the line/link is available. If both users attempt to transmit at the same time, the protocols or the hardware determines who goes first.

control language (CL). The set of all commands with which a user requests system functions.

controller. A device that coordinates and controls the operation of one or more input/output devices (such as work stations) and synchronizes the operation of such devices with the operation of the system as a whole.

controller configuration. The process of creating configuration descriptions for the local (device configuration) and remote (communications configuration) controllers that make up a data processing system. See also *line configuration* and *device configuration*.

controller description. An object that contains a description of the characteristics of a controller that is either directly attached to the system or attached to a communications line.

controlling subsystem. The interactive subsystem that is automatically started first when the system is started and through which the system operator controls the system.

conversation. In interactive communications, the communication between the application program and a specific item (usually another application program) at the remote system.

CTF. See *consumer transaction facility (CTF)*.

CTLD. See *controller description*.

current library. The library that is specified to be the first user library searched for objects requested by a user. The name for the current library can be specified on the Sign-On display or in a user profile. When you specify an object name (such as the name of a file or program) on a command, but do not specify a library name, the system searches the libraries in the system part of the library list, then searches the current library before searching the user part of the library list. The current library is also the library that the system uses when you create a new object, if you do not specify a library name.

data circuit-terminating equipment (DCE). The equipment installed at the user's premises that provides all the functions required to establish, maintain, and end a connection, and the signal conversion and coding between the data terminal equipment and the line. See also *data terminal equipment (DTE)* and *modem*.

data communications. The sending and receiving of data between computers and/or remote devices according to selected protocols.

data description specifications (DDS). A description of the user's database or device files that is entered into the system in a fixed form. The description is then used to create files.

data file. (1) A collection of related data records organized in a specific order. (2) A file created by the specification of FILETYPE(*DATA) on the create commands.

data file utility (DFU). The part of the AS/400 Application Development Tools licensed program that is used to enter, maintain, and display records in a database file.

data link protocol. The physical connection (communications lines, modems, controllers, work stations, and other communications equipment), and the rules (protocols) for sending and receiving data between two or more locations in a data network. Examples of data link protocols include (a) synchronous data link control (SDLC), (b) binary synchronous communications (BSC), (c) asynchronous, (d) X.25, and (e) token-ring network. Contrast with *communications line*.

data management. The part of the operating system that controls the storing and accessing of data to or from an application program. The data can be on internal storage (for example, database), on external media (diskette, tape, or printer), or on another system.

data mode. In data communications, a time during which BSC is sending or receiving characters on the communications line.

data object. A program variable that provides operational and possibly representational characteristics to byte strings in spaces. Contrast with *machine object*.

data queue. An object that is used to communicate and store data used by several programs in a job or between jobs. The system-recognized identifier is *DTAQ.

data stream. All information (data and control commands) sent over a data link usually in a single read or write operation.

data type. (1) A characteristic used for defining data as numeric or character. (2) (C) A set of values together with a set of permitted operations. A data type determines the kind of value that a variable can assume or that a function can return.

database. The collection of all data files stored in the system.

DCE. See *data circuit-terminating equipment (DCE)*.

DDS. See *data description specifications (DDS)*.

deallocate. To release a resource that is assigned to a specific task. Contrast with *allocate*.

decrypt. To convert ciphertext into plaintext. Contrast with *encrypt*.

default. A value automatically supplied or assumed by the system or program.

default printer. A printer that is assigned to a system or user and accepts all the printed output from that system or user, if no other printer is specified.

default program. A user-specified program that is assumed when no other program is specifically named on a debug command, or a user-defined program for handling error messages.

default value. (1) A value supplied by the system that is used when no value is specified by the user. See also *assumed value*. (2) (DDS) The value specified by the user with the DFT keyword in DDS.

definite response. A value in the response-requested field of the request header (RH). The value directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request. Contrast with *exception response*.

delimited identifier. A sequence of one or more characters of the standard character set enclosed within SQL escape characters that are used to form a name.

detail record. A record that contains the daily activities or transactions of a business. For example, the items on a customer order are typically stored in detail records. Contrast with *header record*.

DEVD. See *device description*.

device class. The generic name for a group of device types. For example, all display stations belong to the same device class. Contrast with *device type*.

device configuration. The physical placement of display stations, printers, and so forth; and the configuration descriptions that describe the physical configuration to the system and describe how the configuration will be used by the system. See also *line configuration* and *controller configuration*.

device description. An object that contains information describing a particular device or logical unit that is attached to the system.

device emulation. The programming that allows one device to appear to the user or to a system as another device.

device file. A file that contains a description of how data is to be presented to a program from a device or how data is to be presented to the device from the program. Devices can be display stations, printers, a diskette unit, tape units, or a remote system.

device name. The symbolic name of an individual device.

device type. The generic name for a group of devices. For example, 5219 for IBM 5219 Printers. Contrast with *device class*.

DFU. See *data file utility (DFU)*.

diagnostic. Pertaining to the detection and isolation of an error.

diagnostic message. A message that contains information about errors or possible errors. This message is generally followed by an escape message.

digit. Any of the numerals from 0 through 9.

direct user. A person enrolled in AS/400 Office who is authorized to sign on and use office functions directly. Contrast with *indirect user*.

disk. A direct-access storage medium with magnetically recorded data.

diskette. A thin, removable magnetic disk in a protective jacket.

display screen. The part of the display device, which is similar to a television (TV) picture tube, used to display information entered or received at a work station.

display station. A device that includes a keyboard from which an operator can send information to the system and a display screen on which an operator can see the information sent to or the information received from the system.

download. To send programming instructions from a host system to an attached device. For example, transmitting a type style over a communications line to a 6670 printer.

duplex. Pertains to communications in which data can be sent and received at the same time. Contrast with *half-duplex*.

dynamic. Pertaining to events occurring at run time, or during processing.

EDD. See *Electronic Document Distribution (EDD)*.

EIA-232. In data communications, a specification of the Electronic Industries Association (EIA) that defines the interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) using serial binary data interchange.

Electronic Document Distribution (EDD). An IBM licensed program for the IBM Displaywriter system that supports sending and receiving electronic mail.

emulation. Imitation of one system by another.

emulation program. A control program that permits functions written for one system or device to be run on another system or device.

encrypt. To systematically scramble information so that it cannot be read without knowing the coding key.

end node. A node in an APPN network that can be a source or target node, but does not provide any routing or session services to any other node.

end-of-file delay. An interval during which the system holds a file open after the normal end of the file is reached until one or more records are updated or added to the end of the file. The length of the interval can be specified on the EOFDLY parameter.

end-of-text (ETX) character. The BSC transmission control character used to end a logical set of records that began with the start-of-text character. Contrast with *end-of-transmission-block (ETB) character*.

end-of-transmission (EOT) character. The BSC transmission control character used to end transmission with the remote system.

end-of-transmission-block (ETB) character. The BSC transmission control character used to end a block of records. Contrast with *end-of-text (ETX) character*.

enhanced logical link control (ELLC). An X.25 protocol that allows the transfer of data link control information between two adjoining SNA nodes that are connected through an X.25 packet-switching data network. ELLC enhances error detection and recovery. Contrast with *physical services header (PSH)* and *qualified logical link control (QLLC)*.

Enhanced 5250 Emulation. A feature that allows a personal computer and a printer to be attached to an AS/400 system and perform the functions of one or two 5250 work stations on one twinaxial cable. The work station(s) can be one display, two displays, or one display and one printer.

error log. A record of machine checks, device errors, and media statistics.

exception request. A request that replaces another request in which an error was detected. **Note:** The exception request contains a 4-byte sense field that identifies the error in the original request and, except if there were some path errors, is sent to the destination of the original request; if possible, the sense data is returned in a negative response to the sender of the original request.

exception response. A value in the form-of-response-requested field of a request header. The value requests the receiver to return a response only if the request is unacceptable as received or cannot be processed; that is, only a negative response can be returned. Contrast with *definite response*.

exchange station ID. A control field command and/or response for recognizing the primary station and a secondary station.

exclude authority. An object authority that prevents the user from using the object or its contents. Contrast with *all authority*.

external procedure. A procedure that is not contained within a block. Contrast with *internal procedure*.

externally described data. Data contained in a file for which the fields and the records are described outside of the program (such as with DDS, IDDU, SQL/400), that processes the file. Contrast with *program-described data*.

externally described file. A file in which the records and fields are described to the system when the file is created, and used by the program when the file is processed. Contrast with *program-described file*.

field. A group of related characters (such as name or amount) that are treated as a unit in a record.

field definition. Information that describes the characteristics of data in a field.

FID field. See *format identification (FID) field*.

file. A generic term for the object type that refers to a database file, a device file, or a set of related records treated as a unit. The system-recognized identifier for the object type is *FILE.

file chaining. In AS/400 Query, a function that allows a query application to use data from two database files. The query application views the two chained files as if they were one file and refers to the first file as the primary record format and the second file as the secondary record format.

file definition. (1) (RPG/400) File description and input specifications that describe the records and fields in a file. (2) Information that describes the contents and characteristics of a file.

file overrides. Attributes specified at run time that change the attributes specified in the file description or in the program.

finance communications. The data communications support that allows programs on an AS/400 system to communicate with programs on finance controllers, using the SNA LU session type 0 protocol.

finance device. A device, such as the 4700 Finance Communications System devices and the 3694 Document Processor, that performs functions specifically related to the finance industry. The 3180, 3270, and 5250 work stations are not finance devices.

finance I/O manager (FIOM). (Finance) A set of routines that can be used by an application program to do I/O operations on a finance device that is configured as a non-intersystem communications function (non-ICF) device.

finance support. A part of the system support that uses an AS/400 system as a host to which finance devices can be attached.

FIOM. See *finance I/O manager*.

first speaker. The logical unit (LU) half-session defined when the session is started as the half-session able to begin a bracket without requesting permission from the other LU half-session to do so, and the half-session winning permission if both half-sessions attempt to begin a bracket simultaneously.

fixed currency symbol. A currency symbol that appears in the far left position of an edited field.

fixed length. A specified length for a record or field that cannot be changed.

folder. A directory for documents. A folder is used to group related documents and to find documents by name. The system-recognized identifier for the object type is *FLR. Compare with *library*.

format. A defined arrangement of such things as characters, fields, and lines, usually used for displays, printouts, files, or documents.

format identification (FID) field. A field in each transmission header that indicates the format of the transmission header.

format selector. A user-defined program (either a CL or a high-level language program) that determines where a record should be placed in the database when an application program does not pass a record format name for a record being added to a logical file.

function. (C) A named group of statements that can be called and evaluated, and can return a value to the calling statement.

function key. A keyboard key that allows the user to select keyboard functions or programmer functions. Contrast with *character key*.

function-management-header. A record that contains control information for the data that follows.

global. Pertains to information available to more than one program or subroutine.

graphic character. A character that can be displayed or printed.

half-duplex. Pertaining to data communications that can be sent in only one direction at a time. Contrast with *duplex*.

half-session. One of the locations in a logical connection in a network. See also *session*.

hardware. Physical equipment, rather than programs, procedures, rules, and associated information.

header. In disk management, the 8-byte portion of the 520-byte disk sector used by the operating system for control and access information.

header label. A special set of information on a diskette or tape that describes the contents of the diskette or tape.

header record. A record that contains information, such as customer name and customer address, that is common to detail records. Contrast with *detail record*.

hex. See *hexadecimal*.

hexadecimal. Pertaining to a numbering system with a base of 16.

history log. A summary of the system activities, such as system and job information, device status, system operator messages and a record of program temporary fix (PTF) activity on the system.

I/O. See *input/output*.

ICF. See *intersystem communications function (ICF)*.

ICF file. A device file that allows a program on one system to communicate with a program on another system. There can be one or more sessions with the same or different communications devices at the same time.

identifier. (1) A sequence of bits or characters that identifies a user, program, device, or system to another user, program, device, or system. (2) (COBOL) A data name that is unique or is made unique by the correct combination of qualifiers, subscripts, or indexes. (3) (C) A sequence of letters, digits, and underscores used to identify a data object or function.

image. An electronic representation of an original document recorded by a scanning device.

implicit. Capable of being understood from something else, though unexpressed.

indicator. A 2-character code that is used by a program to test a field or record or to tell when certain operations are to be performed.

indirect user. A person enrolled as an AS/400 Office user who is authorized to handle mail but has no mail log. An indirect user receives printed mail only. Contrast with *direct user*.

initial procedure. An external procedure, started by a calling program, that starts a PL/I program.

initial program. A user-profile program that runs when the user signs on and after the command processor program QCMD is started. QCMD calls the first program.

initial program load (IPL). The process that loads the system programs from the system auxiliary storage, checks the system hardware, and prepares the system for user operations.

initialize. To set the addresses, switches, or the contents of storage to zero, or to the starting value set by the manufacturer.

input field. A field specified in a display file or database file that is used for data the user supplies. Contrast with *output field*.

input stream. A group of records submitted as a batch job that contains CL commands for one or more jobs and data from one or more inline data files.

input/output. Data provided to the computer or data resulting from computer processing.

input/output adapter (IOA). A functional unit or a part of an I/O controller that connects devices to an I/O processor.

input/output controller (IOC). A functional unit that combines the I/O processor and one or more I/O adapters, and directly connects and controls one or more input or output devices.

input/output processor (IOP). A functional unit or the part of an I/O controller that processes programmed instructions and controls one or more input/output devices or adapters.

interactive. Pertaining to the exchange of information between people and a computer. Contrast with *batch*.

interactive job. A job started for a person who signs on to a work station. Contrast with *batch job*.

interface. A shared boundary. An interface might be the hardware to connect two devices or it might be a part of main storage, or registers used by two or more computer programs.

internal procedure. A procedure that is contained within a block. Contrast with *external procedure*.

intersystem communications function (ICF). A function of the operating system that allows a program to communicate interactively with another program or system.

intrasystem communications. A function that allows two programs that are running in two different jobs on the same system to communicate with each other through an ICF file.

IOA. See *input/output adapter (IOA)*.

IOC. See *input/output controller (IOC)*.

IOP. See *input/output processor (IOP)*.

job. A unit of work to be done by a computer.

job name. The name of the job as identified to the system. For an interactive job, the job is assigned the name of the work station at which the job was started; for a batch job, the name is specified in the command used to submit the job.

join logical file. A logical file that combines (in one record format) fields from two or more physical files. See also *logical file*.

key. (COBOL) A data item that identifies the location of a record, or a set of data items that is used to place data in ascending or descending sequence.

keyword. (1) A name that identifies a parameter in a command. (2) (DDS) A name that identifies a function. (3) (RPG/400) A word that is essential to the meaning and structure of a statement in a programming language.

keyword functions. The result of processing DDS keywords in a record format specified on an operation. See also *operation*.

level checking. A function that compares the record format-level identifiers of a file to be opened with the file description that is part of a compiled program to determine if the record format for the file changed since the program was compiled. An error occurs if a record format-level identifier does not exist or does not match the level identifier in the file when a format-level check is done.

level indicator. (1) (COBOL) Two alphabetic characters (FD or SD) that identify the type of file description entry. (2) (RPG/400) Two characters (L0 through L9 and LR) that control calculation and output processing during total time.

library. An object on disk that serves as a directory to other objects. A library groups related objects, and allows the user to find objects by name. Compare with *folder*.

library list. A list that indicates which libraries are to be searched and the order in which they are to be searched. The system-recognized identifier is *LIBL.

library name. A user-defined word that names a library.

licensed program. An IBM-written program that performs functions related to processing user data.

line configuration. The process of creating configuration descriptions for the lines that make up a data processing system. See also *controller configuration* and *device configuration*.

link. (SNA) The combination of the link connection (the transmission medium) and two link stations (one at each end of the link connection). See also *link level*.

link level. (1) (SNA) The combination of the transmission connection, protocol, devices, and programming joining network nodes. (2) (X.25) A part of Recommendation X.25 that defines the link protocol used to get data into and out of the network across the duplex line connecting the subscriber's equipment to the network.

link protocol. The rules for sending and receiving data at the link level.

local. Pertaining to a device or system that is connected directly to or a file that is read directly from your system, without the use of a communications line. Contrast with *remote*.

local address. An address used in a peripheral node in place of a network address and transformed to or from a network address by the boundary function in a subarea node.

local controller. A functional unit within the system that controls the operation of one or more directly attached input/output devices or communications lines. Contrast with *remote controller*.

local location address. (SNA) The address of the logical unit.

local location name. The name by which your system is known to other systems in an SNA network. Equivalent to an SNA local logical unit name. Contrast with *remote location name*.

local system. For interactive jobs, the system to which the display device is directly attached. For batch jobs, the system on which the job is being processed.

local work station. A work station that is connected directly to the system without a need for data transmission functions. Contrast with *remote work station*.

logical file. A description of how data is to be presented to or received from a program. This type of database file contains no data, but it defines record formats for one or more physical files. See also *join logical file*. Contrast with *physical file*.

logical link control. See *enhanced logical link control (ELLC)*, *qualified logical link control (QLLC)*, and *physical services header (PSH)*.

logical unit (LU). One of three types of network addressable units that serve as a port through which a user accesses the communications network. See also *physical unit* and *system services control point (SSCP)*.

LU. See *logical unit (LU)*.

LU-LU session type 0. A type of session between two LU half-sessions using SNA-defined protocols for transmission control and data flow control, but using end-user or product-defined protocols to supplement or replace function management data services protocols. The AS/400 system uses the SNA upline facility support.

LU-LU session type 1. A type of session between an application program and single- or multiple-device data processing display stations in an interactive, batch data transfer, or distributed processing environment. The AS/400 system uses the SNA remove job entry (RJE) support.

LU-LU session type 2. A type of session between an application program and a single display station in an interactive environment, using the SNA 3270 data stream. The AS/400 system uses the 3270 display emulation support.

LU-LU session type 3. A type of session between an application program and a single printer, using the SNA 3270 data stream. The AS/400 system uses the 3270 printer emulation support.

LUSTAT. A command used to send logical unit status information.

machine object. A program object that has no defined storage form; the object is defined internally to the machine. The machine aspect is not available to the user. Contrast with *data object*.

magnetic ink. An ink that contains particles of a magnetic substance whose presence can be detected by magnetic sensors.

magnetic-ink character recognition. The identification of characters by sensing magnetic ink.

manual answer. In data communications, a line type that requires operator actions to receive a call over a switched line. Contrast with *automatic answer*.

manual call. In data communications, a line type requiring operator actions to place a call over a switched line. Contrast with *automatic call*.

mode. The session limits and common characteristics of the sessions associated with advanced-program-to-program communications (APPC) devices managed as a unit with a remote location.

modem. A device that converts data from the computer to a signal that can be sent over a communications line, and converts the communications signal received to data for the computer. See also *data circuit-terminating equipment (DCE)*.

monitor. (1) A functional unit that observes and records selected activities for analysis within a data processing system. (2) Devices or programs that observe, supervise, control, or verify system operations.

multipoint. In data communications, pertains to a network that allows two or more stations to communicate with a single system on one line.

multipoint line. A line or circuit connecting several stations. Contrast with *point-to-point line*.

negative response. In data communications, a reply indicating that data was not received correctly or that a command was incorrect or unacceptable. Contrast with *positive response*. See also *exception response*.

network. A collection of data processing products connected by communications lines for exchanging information between stations.

network file. In object distribution, a file (either a physical file or a save file) sent by one user to one or more other users. A network file is placed on the recipient's message queue when it arrives at the destination system.

network job. In object distribution, a batch input stream sent by one user to one or more users in the network as defined in the system distribution directory.

network message. In object distribution, a message sent by one user to one or more users enrolled in the system distribution directory with the Send Network Message (SNDNETMSG) command.

network node. A node that can define the paths or routes, control route selection, and handle directory services for APPN.

node. (1) One of the systems or devices in a network. (2) A location in a communications network that provides host processing services. (3) (X.25) A point where packets are received, stored, and forwarded to another location (or data terminal equipment) according to a routing method defined for the network.

nonswitched line. A connection between computers or devices that does not have to be made by dialing. Contrast with *switched line*.

normal response mode (NRM). An operational mode of an unbalanced data link in which the secondary station starts transmission only as the result of receiving explicit permission, by polling, from the primary station.

null. The name for an EBCDIC character that represents hex 00. See *null character*.

null character. The character hex 00 used to represent the absence of a displayed or printed character.

object. A named storage space that consists of a set of characteristics that describe itself and, in some cases, data. An object is anything that exists in and occupies space in storage and on which operations can be performed. Some examples of objects are programs, files, libraries, and folders.

operating diskette. (Finance) The diskette that contains the operating image. The operating diskette is used to load the operating image to the main storage of the 4701 Finance Communication Controller.

operating image. (Finance) A collection of the 4701 Finance Communication Controller microcode, your system configuration, and your applications programs.

operating system. A collection of system programs that control the overall operation of a computer system.

Operating System/400 (OS/400). The operating system used by the AS/400 system.

operation. The result of processing statements in a high-level language. See also *keyword functions*.

ordinary identifier. A letter, which may be followed by zero or more characters, each of which is a letter (a-z and A-Z), a symbol ($, @, and #), a number, or the underscore character, used to form a name.

OS. See *operating system*.

OS/400. See *Operating System/400 (OS/400)*.

output. Information or data received from a computer that is shown on a display, printed on the printer, or stored on disk, diskette, or tape.

output field. A field specified in a display file or database file that is reserved for the information processed by a program. Contrast with *input field*.

output/input field. A field specified in a database, display, or ICF file that can be used for both the information supplied to the program and the information received from the program during processing. See also *input field* and *output field*.

parameter. (1) A value supplied to a command or program that is used either as input or controls the actions of the command or program. (2) (COBOL) A variable or a constant that is used to pass values between calling and called programs.

path. (SNA) The set of data links, data link control layers, and path control layers that a path information unit travels through when sent from transmission control of one half-session to transmission control of another half-session.

physical file. A description of how data is to be presented to or received from a program and how data is actually stored in the database. A physical file contains one record format and one or more members. Contrast with *logical file*.

physical services header (PSH). An X.25 protocol used by IBM Systems Network Architecture (SNA) data terminal equipment (DTE). Physical services header provides address services for physically connected systems or devices. Contrast with *enhanced logical link control (ELLC)* and *qualified logical link control (QLLC)*.

physical unit. One of three types of network addressable units. A physical unit exists in each node of an SNA network to manage and monitor the resources (such as attached links and adjacent link stations) of a node, as requested by an system services control point logical unit (SSCP-LU) session.

PLU. See *primary logical unit*.

point-to-point. Pertaining to data transmission between two locations without use of any intermediate terminal or computer.

point-to-point line. A communications line that connects a single remote station to a computer. Contrast with *multipoint line*.

positive response. A response indicating that a request arrived and was successfully received and processed. Contrast with *negative response*. See also *definite response*.

primary logical unit. The logical unit that contains the primary half-session for a particular logical unit to logical unit session. See also *logical unit* and *secondary logical unit*.

problem analysis. The process of finding the cause of a problem. For example, a program error, device error, or user error.

problem log. A record of problems and of the status of the analysis of those problems.

problem management. The part of the systems management function that allows the system to record, analyze, and manage problems and report problems to the provide of service.

procedure. (COBOL) One or more successive paragraphs or sections within the Procedure Division, which direct the computer to perform some action or series of related actions.

processing. The action of performing operations and calculations on data.

processor. A device for processing data from programmed instructions. It may be part of another unit.

program device. A symbolic device that a program uses instead of a real device (identified by the device name). When the program uses a program device, the system redirects the operation to the appropriate real device.

program ID. A one- to eight-character string entered from a finance device and associated with an AS/400 finance transaction program. Lists of valid program IDs and their associated application programs are maintained in program tables.

program mode. The entry mode in which the user can enter BASIC statements and commands into the system from the display station. The formats of the statements are checked as they are entered. Contrast with *data mode*.

program-described data. Data contained in a file for which the fields in the records are described in the program that processes the file. Contrast with *externally described data*.

programmer subsystem. An IBM-supplied interactive subsystem used to code programs on a display station. The system object name is QPGMR.

Programming Request for Price Quotation (PRPQ). A licensed program designed especially for a particular group of customers or an application. Documentation for the program is provided only to those customers who order the PRPQ. Compare with *Request for Price Quotation (RPQ)*.

protocol. A set of rules controlling the communication and transfer of data between two or more devices in a communications system.

PRPQ. See *Programming Request for Price Quotation (PRPQ)*.

pseudocode. A set of instructions that is logically structured but does not follow the syntax of any particular programming language.

PSH. See *physical services header (PSH)*.

PU. See *physical unit*.

public data network. A communication common carrier network that provides data communication services over switched or nonswitched lines.

QCMD. The IBM-supplied control language processor that interprets and processes CL commands for the system.

QLLC. See *qualified logical link control (QLLC)*.

QSRV. The IBM-supplied user profile for the service representative.

qualified logical link control (QLLC). An X.25 protocol that allows the transfer of data link control information between two adjoining SNA nodes that are connected through an X.25 packet-switching data network. The QLLC provides the qualifier "Q" bit in X.25 data packets to identify packets that carry logical link protocol information. Contrast with *enhanced logical link control (ELLC)* and *physical services header (PSH)*.

queue. A list of messages, jobs, or files waiting to be read, processed, printed, or distributed in the order they appear in the list.

quiesce. To become inactive.

random processing. A method of processing in which records can be read from, written to, or deleted from a file order requested by the program that is using them. See also *consecutive processing* and *sequential processing*.

record. A collection of related data or words, treated as a unit; such as one name, address, and telephone number.

recovery. The process of rebuilding databases after a system failure.

release-program-device operation. An operation that makes a program device not available for input/output operations. Contrast with *acquire-program-device operation*.

remote. Pertaining to a device, system, or file that is connected to another device, system, or file through a communications line. Contrast with *local*.

remote controller. A device or system, attached to a communications line, that controls the operation of one or more remote devices. Contrast with *local controller*.

remote device. A device whose controller is connected to an AS/400 system by a communications line.

remote location name. Any other system with which your system can communicate in an SNA network. This corresponds to the remote location name specified in the communications configuration. Contrast with *local location name*.

remote work station. A work station that is connected to the system by data communications. Contrast with *logical work station*.

Request for Price Quotation (RPQ). A customer request for a price quotation on alterations or additions to the functional capabilities of a computing system hardware product, or device. Compare with *Programming Request for Price Quotation (PRPQ)*.

request unit (RU). The record transmitted to the other system. This record can contain a request, data, or both. Contrast with *response unit (RU)*.

request/response header (RH). Control information preceding a request/response unit that specifies the type of request/response unit and contains control information associated with that request/response unit. See also *request unit (RU)*.

request/response unit (RU). A combined term to identify a request unit or a response unit.

response header (RH). A header, optionally followed by a response unit, that indicates whether the response is positive or negative and that may contain a pacing response. See *request/response header*. Contrast with *request header*. See also *negative response* and *positive response*.

response indicator. A 1-character field 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.

response unit (RU). The record sent to respond to a request. The response can be either positive or negative and can include control information. Contrast with *request unit (RU)*.

restore. To copy data from tape, diskette, or a save file to auxiliary storage. Contrast with *save*.

return code. In data communications, a value sent by the system to a program to indicate the results of an operation by that program.

return indicator. An indicator to an RPG/400 program that control should be returned to the calling program.

RH. See *request header* or *response header*.

routine. A set of statements in a program that causes the system to perform an operation or a series of related operations.

RPG. Report Program Generator. A programming language designed for writing application programs for business data processing requirements. The application programs range from report writing and inquiry programs to applications such as payroll, order entry, and production planning.

RPG/400. An IBM licensed program that is the SAA RPG programming language available on the AS/400 system, including system-specific functions.

RPQ. See *Request for Price Quotation (RPQ)*.

RSHUTD. Command used to request an orderly end to a session.

RU. See *request unit* or *response unit (RU)*.

RU chain. A set of related request/response units that are transmitted consecutively on a particular normal or expedited data flow. See also *bracket*.

save. To copy specific objects, libraries, or data by transferring them from main or auxiliary storage to magnetic media such as tape, diskettes, or a save file. Contrast with *restore*.

SDLC. See *synchronous data link control (SDLC)*.

SDT. A command issued by the primary logical unit, which allows user data to be sent on the logical unit-to-logical unit (LU-to-LU) session.

secondary logical unit (SLU). The logical unit that contains the secondary half-session for one logical unit-to-logical unit (LU-to-LU) session. See also *logical unit (LU)*. Contrast with *primary logical unit (PLU)*.

secure. Controlling who can use and to what extent an object can be used by controlling the authority given to the user.

security officer. A person assigned to control all of the security authorizations provided with the system. A security officer can, for example, remove password or resource security; or add, change, or remove security information about any system user.

sense data. Data sent with a negative response, indicating the reason for the response.

sequential processing. A method of processing in which records are read, written to, or deleted in the order determined by the value of the key field. See also *consecutive processing* and *random processing*.

service processor. The logic that contains the processor function to start the system processor and handle error conditions. See also *system processor*.

session. (Finance) A logical connection by which an AS/400 system can communicate with a finance controller.

session default. A session assumed by the system for temporary use.

severity code. A number that indicates how important a message is. The higher the number, the more serious the condition.

SLU. See *secondary logical unit (SLU)*.

SNA. See *Systems Network Architecture (SNA)*.

SNA network. The part of the user application network that conforms to the formats and protocols of Systems Network Architecture. The SNA network consists of network addressable units, boundary function parts, and the path control network.

SNBU. See *Switched Network Backup*.

SSCP. See *system services control point (SSCP)*.

start-of-header (SOH) character. In binary synchronous communications, the transmission control character indicating that the information that follows is a header.

start-of-text (STX) character. In binary synchronous communications, a transmission control character used to begin a logical set of records that will be ended by the end-of-text character or end-of-transmission-block character.

store. To put or keep data in a storage device.

subroutine. (1) A group of instructions within another group of instructions that can be called by another program or another subroutine. (2) In data communications, a group of statements in a program that can be run several times in that program. (3) (RPG/400) A group of calculation specification statements in a program that can be run several times in that program.

subsystem. An operating environment, defined by a subsystem description, where the system coordinates processing and resources.

subsystem description. A system object that contains information defining the characteristics of an operating environment controlled by the system.

switched line. In data communications, a connection between computers or devices that is established by dialing. Contrast with *nonswitched line*.

Switched Network Backup. A feature of the modem that allows a nonswitched line to be used alternatively as a switched line or allows a switched line to be used as a nonswitched line depending on the characteristics of the modem. Abbreviated SNBU.

synchronous data link control (SDLC). (1) A form of communications line control that uses commands to control the transfer of data over a communications line. (2) A communications discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-Level Data Link Control (HDLC) of the International Standards Organization (ISO), for transferring synchronous, code-transparent, serial-by-bit information over a communications line. Transmission exchanges may be duplex or half-duplex over switched or nonswitched lines. The configuration of the connection may be point-to-point, multipoint, or loop. Compare with *binary synchronous communications (BSC)*.

synchronous level. A level at which a logical unit (LU) determines if it can allocate and deallocate system resources.

synchronous processing. A series of operations that are done as part of the job in which they were requested; for example, calling a program in an interactive job at a work station.

synchronous transmission. A method of transmission in which the sending and receiving of data is controlled by timing signals.

syntax. The rules for constructing a command or statement.

system library. The library shipped with the system that contains objects, such as authorization lists and device descriptions created by a user; and the licensed programs, system commands, and any other system objects shipped with the system. The system identifier is QSYS.

system monitor. (Finance) A 4700 controller program used to perform service, configuration, and debugging functions on that controller.

system name. (1) An IBM-supplied name that uniquely identifies the system. It is used as a network value for certain communications applications such as APPC. (2) An IBM-defined name that has a predefined meaning to the COBOL compiler. System names include computer names, language names, device names, and function names.

system object. One of two machine object classifications. Any of the machine objects shipped with the system or any of the operating system objects created by the system.

system processor. The logic that contains the processor function to translate and process the OS/400 control language commands and programming language statements. See also *service processor*.

system resources. Those items controlled by the system, such as programs, devices, and storage areas that are assigned for use in jobs.

system security. A system function that restricts the use of files, libraries, folders, and devices to certain users.

system services control point (SSCP). A focal point within an SNA network for managing the other systems and devices, coordinating network operator requests and problem analysis requests, and providing directory routing and other session services for network users.

system value. Control information for the operation of certain parts of the system. A user can change the system value to define his working environment. System date and library list are examples of system values.

System/36 environment. A function of the operating system that processes most of the System/36 operator control language (OCL) statements and procedure statements to run System/36 application programs and allows the user to process the control language (CL) commands. Contrast with *System/38 environment*.

System/38 environment. A function of the operating system that processes most of the System/38 control language (CL) statements and programs to run System/38 application programs. Contrast with *System/36 environment*.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences that are used for transmitting information units through networks, as well as controlling the configuration and operation of networks.

table. (1) (RPG/400) A series of elements with like characteristics. A table can be searched for a uniquely identified element, but elements in a table cannot be accessed by their position relative to other elements. (2) (COBOL) A set of logically consecutive data items that are defined in the Data Division with the OCCURS clause.

target. In advanced program-to-program communications, the program or system to which a request for processing is sent.

transaction. In communications, an exchange between a program on a local system and a program on a remote system that accomplishes a particular action or result. See also *conversation* and *session*.

transaction program. A user-supplied application program for processing data received by the AS/400 system from a finance device.

UDDS. See *user-defined data stream*.

unbind command. A command used to reset the protocols for a session. Contrast with *bind command*.

upload. To send programming instructions to a host system from an attached device.

user ID. See *user identification (user ID)*.

user ID/address. The two-part network name used in the system distribution directory and in the office applications to uniquely identify a user and send electronic mail.

user identification (user ID). The name used to associate the user profile with a user when a user signs on the system. See also *user profile name*.

user password. A unique string of characters that a system user must enter to identify himself to the system, if the system resources are secured.

user profile. An object with a unique name that contains the user's password, the list of special authorities assigned to a user, and the objects the user owns.

user profile name. The name or code that the system associates with a user when he or she signs on the system. Also known as user ID. See also *user identification (user ID)*.

user-defined data stream (UDDS). (Finance) A data stream in which the user has defined and embedded all device control characters.

vary off. To make a device, controller, or line unavailable for its normal, intended use.

vary on. To make a device, controller, or line available for its normal, intended use.

X.21. In data communications, a specification of the CCITT that defines the connection of data terminal equipment to an X.21 (public data) network.

X.25. In data communications, a specification of the CCITT that defines the interface to an X.25 (packet-switching) network.

5250 emulation. Any one of many licensed programs that allows a personal computer to perform like a 5250 display station, and use the functions of an AS/400 system.

Index

D

E

F

R

S

T

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your IBM-approved remarketer. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

☐ If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your IBM representative or your IBM-approved remarketer to request additional publications.

Name _____

Company or Organization _____

Address _____

_____

City State Zip Code

Phone No. _____

Area Code

No postage necessary if mailed in the U.S.A.

Cut or F
Along L

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

International Business Machines Corporation
Information Development
Department 245
3605 North Hwy 52
ROCHESTER MN 55901-9986

Fold and Tape | Please do not staple | Fold and Tape

IBM®

Cut or Fol
Along Lin

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your IBM-approved remarketer. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

☐ If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your IBM representative or your IBM-approved remarketer to request additional publications.

Name _____

Company or Organization _____

Address _____

_____

City State Zip Code

Phone No. _____

Area Code

No postage necessary if mailed in the U.S.A.

Cut or F
Along L

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

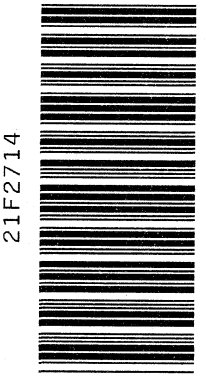
International Business Machines Corporation
Information Development
Department 245
3605 North Hwy 52
ROCHESTER MN 55901-9986

Fold and Tape | **Please do not staple** | Fold and Tape

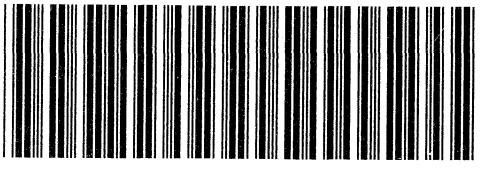
IBM®

Cut or Fc
Along Lii

Program Number
5728-SS1



SC21-8099-1



Printed in U.S.A.